

René Langermann

**Beitrag zur durchgängigen Simulationsunterstützung im
Entwicklungsprozess von Flugzeugsystemen**

**Braunschweig : Institut für Luft- und Raumfahrtsysteme,
2009**

Zugl.: Braunschweig, Technische Universität, Diss., 2008

Veröffentlicht: 02.04.2009

<http://www.digibib.tu-bs.de/?docid=00026819>

Beitrag zur durchgängigen Simulationsunterstützung im Entwicklungsprozess von Flugzeugsystemen

Von der Fakultät für Maschinenbau
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte Dissertation

von: René Langermann

aus: Winsen (Luhe)

eingereicht am: 09.06.2008

mündliche Prüfung am: 09.09.2008

Vorsitzender:	Prof. Dr.-Ing. Ferit Küçükay
Referenten:	Prof. Dr.-Ing. Peter Vörsmann
Referenten:	Prof. Dr.-Ing. Ulrich Seiffert

2008

Vorwort

Die vorliegende Dissertation entstand im Rahmen meiner Tätigkeit als Mitarbeiter der EADS Innovation Works in Hamburg. Viele der Ergebnisse entstanden während meiner Mitarbeit an dem nationalen Forschungsprojekt „Advanced Power and Water Generation System“ (APAWAGS) und dem europäischen Forschungsprojekt „Fuel Cell Application in a new configured Aircraft“ (CELINA).

Mein besonderer Dank gilt meinem Doktorvater Professor Dr.-Ing. Peter Vörsmann für sein Vertrauen und die sehr gute Zusammenarbeit zwischen dem Institut für Luft- und Raumfahrtssysteme in Braunschweig und der EADS IW in Hamburg. Sein technischer Rat und nicht zuletzt seine moralische Unterstützung haben zum Gelingen dieser Arbeit erheblich beigetragen.

Professor Dr.-Ing. Ullrich Seiffert danke ich für die Übernahme des Korreferats. Seine inhaltlichen Anregungen und Kritiken, insbesondere im Bereich der Automobilindustrie und Brennstoffzellentechnologie, haben diese Arbeit bereichert.

Meinem Arbeitgeber, der EADS Deutschland GmbH, danke ich für die Möglichkeit eines industrienahen und praxisorientierten Dissertationsthemas. Meinen Kollegen danke ich für die tolle Teamarbeit und das einzigartige Betriebsklima. Mein besonderer Dank gilt Herrn Dipl.-Ing. Jörg Christoffers, der mich in zahllosen Diskussionen und konstruktiven Gesprächen während der gesamten Promotion unterstützt hat.

Des Weiteren möchte ich mich an dieser Stelle bei meinen studentischen Hilfskräften Frau Dipl.-Ing. Wiggers, Herr Dipl.-Ing. Schramm, Herr Dipl.-Ing. Lancelle, Herr Dipl.-Ing. Ullmann, Herr Dipl.-Ing. Cwielong, Herr Dipl.-Ing. Richter und Herr Dipl.-Ing. Lauckner bedanken, die mich während der vierjährigen Promotionszeit unterstützt haben.

Frau Semenowitsz danke ich für die Korrekturlesung sowie für ihre formalen Verbesserungsvorschläge. Meiner Frau und meinen Kindern danke ich dafür, dass sie mich immer unterstützt haben.

Hamburg, 6. 6.2008

René Langermann

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Motivation	1
1.2	Kapitelübersicht	2
2	Simulation in der Systementwicklung.....	3
2.1	Grundlagen der Simulationstechnik.....	3
2.2	Systementwicklungsprozess	6
2.2.1	Allgemeine Prozessmodelle	6
2.2.2	Flugzeugsystementwicklungsprozess	10
2.3	Einsatz von Simulation im Entwicklungsprozess.....	11
2.3.1	Modelldetaillierung im Entwicklungsprozess	13
2.3.2	Modellierung und Simulation multidisziplinärer Systeme	14
2.3.3	Einsatz von Simulation in der Automobilindustrie.....	17
2.3.4	Einsatz von Simulation in der Luftfahrt	21
2.4	Informationsbereitstellung	22
2.4.1	Anforderungen an die Informationsbereitstellung in der Produktentwicklung ...	22
2.4.2	Möglichkeiten der Informationsbereitstellung	23
3	Analyse des Einsatzes von Simulation im Entwicklungsprozess von Flugzeugsystemen	26
3.1	Istsituation der Systementwicklung	26
3.2	Analyse der Systementwicklung in der Flugzeugindustrie	28
3.2.1	Informationsaustausch.....	28
3.2.2	Simulationseinsatz in der Entwicklung von Flugzeugsystemen	30
3.3	Zusammenfassung der Defizite im bestehenden Entwicklungsprozess	34
3.4	Anforderungen	37
4	Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation.....	43
4.1	Simulationskonzept	44
4.1.1	Simulationskonzeptalternativen	44
4.1.2	Detaillierung des Simulationskonzeptes	48
4.1.3	Zusammenfassung des Simulationskonzeptes	57
4.2	Informationsmanagement.....	60
4.2.1	Alternativen für das Informationsmanagement.....	60
4.2.2	Detaillierung des Informationsmanagements	64
4.3	Standardisierte Modellierung.....	76
4.4	Gesamtkonzept.....	80
4.4.1	Konzeptphase (M4–M5).....	80
4.4.2	Detaillierungsphase (M5–M7).....	81
4.4.3	Komponentenentwicklung (M7–M8)	81
4.4.4	Testphase (M8–M11).....	82
5	Gesamtkonzeptevaluierung.....	84
5.1	Grundlagen von Brennstoffzellensystemen.....	84
5.2	Realisierung des Simulationskonzepts.....	95
5.2.1	Architekturmodell des Brennstoffzellensystems	95
5.2.2	Funktionales Modell der Brennstoffzelle.....	107
5.2.3	Gestaltorientierte Simulation eines Brennstoffzellensystems.....	110
5.2.4	Verteilte Simulation des Brennstoffzellensystems.....	119
5.3	Informationsmanagement.....	125
5.3.1	Standardisierte Bereitstellung von Simulationsdaten	126
6	Zusammenfassung und Ausblick	137
7	Anhang	139
7.1	Mathematische Grundlagen der Simulationstechnik	139
7.2	Gestaltorientierte Simulation	144
7.3	Systemsimulation.....	148
7.3.1	Mathematisch orientierte Simulationsprogramme	148
7.3.2	Signalfussorientierte Simulationsprogramme	149
7.3.3	Symbolorientierte Simulationsprogramme.....	150
7.4	Datenaustauschformate und Informationsbereitstellung.....	154
7.4.1	Softwareprogramme zur Simulationsdatenverwaltung	154
7.4.2	Verfügbare Simulationsdatenformate	156

Inhaltsverzeichnis

7.5	Prozessanalyse mit ARIS	159
7.6	Modellierung mit SysML	161
7.7	Standardisierte Schnittstelle	163
Literaturverzeichnis.....		164
Stichwortverzeichnis		178

Abkürzungsverzeichnis

A/C	Aircraft
ABS	Anti-Blockier-System
AC	Alternating Current
AG	Aktiengesellschaft
AMDB	Airbus Modelling Database
AMO	Aircraft Models Management
AMS	Analog/Mixed Signal
API	Application Programming Interface
APU	Auxiliary Power Unit
ARP	Aerospace Recommended Practice
ASM	Air Separation Module
ATA	Air Transportation Association
ATR	Autothermal Reformer
BDF	Rückwärtsdifferenzen-Verfahren
BLAS	Basic Linear Algebra Subprograms
BMW	Bayerische Motoren Werke
BZ	Brennstoffzelle
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CATIA	Computer Aided Three-Dimensional Interactive Application
CAX	Computer Aided x
CFD	Computational Fluid Dynamics
CORBA	Common Object Request Broker Architecture
CSV	Comma Separated Values
CVS	Concurrent Versions System
DAE	Differential Algebraic Equation
DB	Datenbank
DBMS	Datenbankmanagementsystem
DC	Direct Current
DGL	Differenzialgleichungen
DMSO	Defense Modeling and Simulation Office
DMU	Digital Mock-Up
DOC	Direct Operating Costs
DoE	Design of Experiments
DOM	Document Object Model
EADS	European Aeronautic, Defense and Space Company
EDM	Engineering Data Management
EDMS	Engineering Data Management System
EG	Europäische Gemeinschaft
EISPACK	Eigensystem-Package
ERM	Entity-Relationship-Modell
ESP	Elektronisches Stabilitätsprogramm
EU	Europäische Union
FC	Fuel Cell
FDM	Finite-Differenzen-Methode
FedEX	Federation Execution
FEM	Finite-Elemente-Methode
FSEP	Flugzeugsystementwicklungsprozess
FVM	Finite-Volumen-Methode
GPL	General Public License
GTL	Gas-to-Liquid
GUI	Graphical User Interface
HDF	Hierarchical Data Format
HIL	Hardware in the Loop
HLA	High Level Architecture
HSAD	Hauptsystem-Anforderungsdokument
HT	High Temperature
HTML	Hypertext Markup Language

Abkürzungsverzeichnis

HTTP	Hypertext Transfer Protocol
HW	Hardware
IB	Informationsbasis
ICD	Interface Control Document
IEEE	Institute of Electrical and Electronics Engineering
INCOSE	International Council on Systems Engineering
IP	Internet Protocol
ISA	International Standard Atmosphere
ISAR	Integrierte Simulationsumgebung für Fahrdynamik mit Regelsystem
ISO	International Organization for Standardization
IT	Informationstechnologie
IW	Innovation Works
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
KCL	Kirchoff's Current Law
LAN	Local Area Network
LINPACK	Linear Algebra Package
LTS	Niedertemperatur-Shift
M	Meilenstein
MATLAB	Matrix Laboratory
MEA	More Electrical Aircraft
MIL	Model in the Loop
MOF	Meta Object Facility
MTS	Mitteltemperatur-Shift
NASA	National Aeronautics and Space Administration
NCSA	National Center for Supercomputing Applications
ODE	Ordinary Differential Equation
OMG	Object Management Group
OPEC	Organisation of Petroleum Exchanging Countries
PDF	Portable Document Format
PDh	Partielle Dehydrogenierung
PDM	Production Data Management
PEMFC	Proton Exchange Membrane Fuel Cell
Pkw	Personenkraftwagen
POX	Partielle Oxidation
PPS	Produktplanungs- und Steuerungssystem
PROX	Preferential Oxidation
PTS	Purchaser Technical Specification
RANSE	Reynolds Averaged Navier Stokes Equations
RAT	Ram Air Turbine
RTI	Run-Time Infrastructure
RTIExec	RTI Execution
SAD	System-Anforderungsdokument
SAE	Engineering Society for Advancing Mobility Land Sea Air and Space
SE	Simultaneous Engineering
SIL	Software in the Loop
SimDMS	Simulation-Datenmanagementsystem
SOFC	Solid Oxide Fuel Cell
SPICE	Simulation Program with Integrated Circuit Emphasis
SQL	Structured Query Language
SRD	System Requirement Document
SRML	Simulation Reference Markup Language
STEP	Standard for Exchange of Product Model Data
STR	Steam Reforming
SW	Software
SysML	System Modeling Language
TCP	Transmission Control Protocol
TLS	Transport Layer Protocol
TLSRD	Top Level System Requirements Document
TSD	Technisches Spezifikationsdokument

Abkürzungsverzeichnis

UDP	User Datagram Protocol
UML	Unified Modeling Language
V&V	Verification and Validation
VDI	Verein Deutscher Ingenieure
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VPM	Virtual Product Management
VV&A	Verification, Validation and Accreditation
W3C	World Wide Web Consortium
XMI	XML Metadata Interchange
XML	Extensible Markup Language
XSD	XML-Schema-Definition
XT	Extreme Tailoring

Formelzeichen

Symbol	Einheit	Physikalische Größe
A	m ²	Fläche
A	-	Koeffizient der Tafelgleichung
C _p	J/K	Wärmekapazität
c	N/m	Federkonstante
d	kg/s	Dämpfung
E	J	Energie
F	C/mol	Faraday-Konstante
\vec{f}	m/s ²	Beschleunigungsvektor
G	J	Gibbssche Freie Enthalpie
H	J	Enthalpie
H _U	J/kg	spezifischer Heizwert
Δh _R	kJ/mol	Standardbildungsenthalpie
I	A	Strom
i	A/m ²	Stromdichte
i ₀	A/m ²	Austauschstromdichte
J	-	Jacobi-Matrix
K	-	Gewichtungskoeffizient
m	-	Koeffizient der Massentransfer-Gleichung
m	kg	Masse
n	-	Koeffizient der Massentransfer-Gleichung
n	-	Zellenanzahl
\dot{n}	mol/s	Molstrom
O ₂ /C	-	Sauerstoff-zu-Kohlenstoffverhältniss
P	W	Leistung
p	Pa	Druck
Q	J	Wärme
\vec{q}	J kg ⁻¹ K ⁻¹	Wärmestromvektor
R	J kg ⁻¹ K ⁻¹	Gaskonstante
r	Ω m ²	Flächenspezifischer Widerstand
S	J/K	Entropie
S/C	-	Wasserdampf-zu-Kohlenstoffverhältniss
T	K	Temperatur
T	s	Zeitkonstante
t	s	Zeit
U	V	Spannung
u	m/s	Geschwindigkeit in x-Richtung
v	m/s	Geschwindigkeit in y-Richtung
\vec{v}	m/s	Geschwindigkeitsvektor
w	m/s	Geschwindigkeit in z-Richtung
x	-	Dampfanteil

Griechische Formelzeichen

Symbol	Einheit	Physikalische Größe
η	-	Wirkungsgrad
λ	-	Luftüberschusszahl
ρ	kg/m^3	Dichte
$\bar{\tau}$	N/m^2	Spannungstensor
Φ	-	Dissipationsfunktion
Φ	-	relative Feuchtigkeit

Indizes

A	Nachbrenner
EL	elektrisch
IN	eintretend
m	molar, stoffmengenbezogen
N	Nernst-Leerlaufspannung
n	molar, stoffmengenbezogen
OUT	austretend
PER	Peripherie
rev	reversibel
SYS	System
Zelle	eine Brennstoffzelle

1 Einleitung

Es liegt in der Natur des Menschen, Fragen über seine Zukunft zu stellen. So baten bereits im dritten Jahrhundert vor Christus die großen Lenker ihrer Zeit das Orakel von Delphi um Rat. Die Orakelsprüche der von einem Priesterkollegium unterstützten Pythia waren für viele wichtige Entscheidungen der damaligen Zeit ausschlaggebend. Ihre Weissagungen wurden von den anwesenden Priestern interpretiert und basierten auf dem Informationsnetz und damit auf dem umfangreichen Wissen der Griechen. „Das Wissen und die Vision sind die Basis für Weissagungen in die Zukunft“ [137].

Auf anderen Kontinenten stellten die Menschen ähnliche Fragen über ihrer Zukunft. So berechneten im heutigen Mexiko die Azteken bereits vor 600 Jahren die Bahnen der Planeten und Sternkonstellationen. Die Umlaufzeiten der damals fünf sichtbaren Planeten bestimmten sie teilweise bis auf wenige Minuten genau. Sie glaubten daran, dass bestimmte kosmische Ereignisse Einfluss auf ihr Leben bzw. eine spirituelle Bedeutung hatten. Die Vorausberechnung der Planetenbahnen kann bereits als Simulation bezeichnet werden, da ein empirisches Modell des Sternenhimmels als Basis für die Vorhersage benutzt wurde.

Die Frage nach der Zukunft ist heute so aktuell wie zur damaligen Zeit. Allerdings haben sich die Methoden zur Vorhersage zukünftiger Ereignisse grundlegend verändert. Heutzutage bietet die Computertechnologie eine effiziente Möglichkeit der Prediktion. Die stetige Weiterentwicklung dieser Technologie erlaubt es, immer komplexere Zusammenhänge zu simulieren. Mit Wettersimulationen versuchen Meteorologen zum Beispiel, das Wetter der nächsten Tage vorherzubestimmen. Dass sie dabei nicht immer richtig liegen, ist offensichtlich, dennoch hat sich die Wahrscheinlichkeit des Eintretens des prognostizierten Wetters in den vergangenen zwei Jahrzehnten deutlich verbessert. Für den Ingenieur bietet die Computersimulation ein geeignetes Hilfsmittel zur Vorausbestimmung von Systemverhalten. Aus diesem Grund hat sich der Einsatz der computergestützten Simulation in nahezu allen technischen Disziplinen etabliert.

1.1 Motivation

Für technische Problemstellungen jeder Art existieren heute unzählige spezialisierte Simulationsprogramme, mit deren Hilfe einzelne physikalische Disziplinen modelliert und simuliert werden können. Allerdings wurden bisher für jedes zu simulierende Problem einzelne, in sich geschlossene und nicht wiederverwendbare Simulationsmodelle erstellt [106]. Für den Entwicklungsprozess bedeutet das, dass die an der Produktentwicklung beteiligten Fachdisziplinen Insellösungen generieren. Der Blick für das Gesamtsystem geht verloren, da jeweils nur in Teillösungen gedacht wird. Eine optimale Gesamtsystemlösung wird sich allerdings niemals aus der Summe der optimalen Teilsystemlösungen ergeben. Für die Entwicklung qualitativ hochwertiger Produkte ist das ganzheitliche Denken und damit das Verständnis für die Interaktionen der Systemkomponenten von entscheidender Bedeutung. Der durchgängige Einsatz von Simulationen in allen Produktentwicklungsphasen verbessert das Gesamtsystemverständnis und ist heute für die Kostenreduzierung sowie die Steigerung der Qualität und damit der Wettbewerbsfähigkeit eines Unternehmens von entscheidender Bedeutung [140], [9], [91].

Für die erfolgreiche Entwicklung eines Produktes spielen die frühen Entwicklungsphasen eine übergeordnete Rolle, da in diesen Phasen ein Großteil der Entwicklungskosten festgelegt wird [55], [145], [13], [86], [84]. Aus diesem Grund ist es bereits zu Beginn der Entwicklung von elementarer Bedeutung, nicht die nächstbeste Lösung, sondern die beste Lösung zu ermitteln [175]. Durch den Einsatz der Simulationstechnik ist es möglich, Konzeptalternativen auf einer mathematischen Basis miteinander zu vergleichen und so die Suche nach der besten Lösung zu unterstützen. In der Praxis bietet sich allerdings ein anderes Bild. Aufgrund von Ressourcenknappheit wird in den Phasen der frühen Produktentwicklung häufig auf den Einsatz von Simulationstechniken verzichtet [174]. Aus dem Lösungsraum der möglichen Produktalternativen wird ein Konzept ausgewählt, welches in den seltensten Fällen mit der besten Lösung übereinstimmt. Zusätzlich werden die existierenden Modelle nur ungenügend weiterverwendet. Zurückzuführen ist dieses einerseits auf die in sich geschlossenen Simulationsmodelle der verschiedenen Fachdisziplinen und andererseits auf die unzureichende Verwaltung der Simulationsdaten in PDM-Systemen (engl.: Product Data Management) [108].

1.1 Motivation

Ein weiteres Problem besteht darin, dass für den Einsatz von Simulation im Entwicklungsprozess keine standardisierten Vorgehensweisen, Methodiken oder Konzepte existieren. Zusätzlich wird die interdisziplinäre Kommunikation der beteiligten Entwicklungsteams durch unterschiedliche Begriffswelten erschwert [86].

Die Durchführung einer Simulationsstudie ist wesentlich vom Datenmaterial, welches für die Erstellung der Modelle und die Eingabe der Simulationsparameter eingesetzt wird, abhängig. Allerdings stehen die benötigten Daten oftmals nicht zur Verfügung oder müssen mit erheblichem Aufwand zunächst gesucht werden. Merk schreibt, dass 51 % der Gesamtzeit bei der Durchführung einer Simulationsstudie für die Datenbeschaffung benötigt werden [108]. Damit ist der Zeitanteil für die Datenbeschaffung im Vergleich zur eigentlichen wertschöpfenden Tätigkeit sehr hoch. Oftmals werden Informationen sogar bewusst zurückgehalten, da ein Wissensverlust befürchtet wird. In einem Entwicklungsprozess, in dem alle Beteiligten dasselbe Ziel verfolgen – den Erfolg des Gesamtprodukts –, ist diese Denkweise äußerst schädlich, da sie der Verbesserung des ganzheitlichen Denkens entgegenwirkt. „Wissen ist die einzige Ressource, die sich in ihrem Gebrauch nicht verzehrt, sondern vermehrt“ [74].

Ein weiteres Problem besteht darin, dass Simulationsmodelle oftmals nicht in der jeweils für den Anwendungsfall benötigten Detaillierung vorliegen. Simulationsmodelle in der frühen Entwicklungsphase können z. B. sehr einfache Systemzusammenhänge wiedergeben; sie dienen lediglich zum Vergleich von Konzeptalternativen. Hingegen werden im weiteren Verlauf der Entwicklung immer detailliertere Modelle benötigt. Nach Einstein heißt es: „Mache die Dinge so einfach wie möglich, aber nicht einfacher“ [24].

Zusammenfassend lässt sich sagen, dass kein standardisiertes Vorgehen, Konzept und keine Methode zum durchgängigen Einsatz von Simulation in der Systementwicklung existiert. Bedingt dadurch, kommt es in der Entwicklung zu Verzögerungen und kostenintensiven Iterationen. Ziel dieser Arbeit ist die Entwicklung eines Konzeptes, welches den Einsatz von Simulation im Entwicklungsprozess vorgibt und somit zur Steigerung der Qualität und Transparenz, Reduzierung der Entwicklungszeit und -kosten sowie zur Verbesserung des ganzheitlichen Denkens im Produktentstehungsprozess führt. „Die Frage ist nicht, ob, sondern wie dieses wertvolle Werkzeug in die tägliche Arbeit zu integrieren ist“ [57].

1.2 Kapitelübersicht

In dieser Arbeit werden bereits bekannte Methodiken, Prozesse sowie Technologien untersucht, die als Basis eines Konzeptes für die durchgängige Simulationsunterstützung im Systementwicklungsprozess von Flugzeugen zu fungieren. Es wird gezeigt, wie diese Methodiken in einem Gesamtkonzept in einer realen Entwicklung umgesetzt werden können. Zunächst werden in Kapitel zwei die Grundlagen zum Thema Simulation und Entwicklungsprozess bereitgestellt. Um eventuell Vorgehensweisen anderer Industriezweige zu übernehmen und an die Anforderungen der Flugzeugindustrie anzupassen, wurde der Simulationseinsatz in der Automobilindustrie untersucht. Im dritten Kapitel wird der derzeitige Einsatz von Simulationstechniken in der Luftfahrt analysiert. Basierend auf den Defiziten der Istanalyse werden Anforderungen an ein zukünftiges Konzept für den durchgängigen Simulationseinsatz abgeleitet. Die Anforderungen lassen sich in die Gruppen Simulation, Modellierung und Informationsbereitstellung aufteilen. Diese drei Hauptanforderungen dienen in Kapitel vier als Hauptpfeiler des zukünftigen Konzeptes, für die jeweils verschiedene Einzelkonzepte analysiert und bewertet werden. Anschließend werden die Einzelkonzepte anhand der Anforderungen aus Kapitel drei validiert. Abgeschlossen wird das vierte Kapitel mit der Erstellung des Gesamtkonzeptes, bestehend aus den Einzelkonzepten für die Simulation, die Modellierung und die Informationsbereitstellung. Im fünften Kapitel erfolgt die Umsetzung des Konzeptes anhand einer Brennstoffzellensystementwicklung. Abgeschlossen wird die Arbeit mit einer Zusammenfassung der Ergebnisse sowie einem Ausblick.

2 Simulation in der Systementwicklung

In diesem Kapitel werden der Stand der Simulationstechnologie und ihr Einsatz in verschiedenen Industriezweigen beschrieben. Zunächst werden die verschiedenen Simulationstechniken erläutert und klassifiziert. Anschließend werden allgemeine Prozessmodelle zur Systementwicklung vorgestellt, die für den Flugzeugsystementwicklungsprozess von Interesse sind. Abgeschlossen wird dieses Kapitel mit der Bereitstellung von Informationen im Entwicklungsprozess.

2.1 Grundlagen der Simulationstechnik

Nach der VDI-Richtlinie 3633 ist Simulation folgendermaßen definiert [177]:

„Simulation ist ein Verfahren zur Nachbildung eines Systems mit seinen dynamischen Prozessen mithilfe eines experimentierfähigen Modells, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.“

Das Ziel jeder Simulation ist das Erlangen von Erkenntnissen basierend auf dem Systemmodell, die auf das reale System übertragen werden können [42]. Dabei können Simulationen sehr unterschiedlich ausfallen. So eignen sich physikalische Nachbildungen des realen Systems genauso zur Simulation wie mathematische Modelle. Die physikalische Nachbildung weist eine Reihe von Nachteilen gegenüber der mathematisch basierten Simulationstechnik auf, sie erfordert z. B. kostenintensive Experimente, die wenig flexibel sind. Aus diesem Grund liegt der Fokus dieser Arbeit auf der mathematisch basierten Simulation.

Mathematische Methoden lassen sich prinzipiell in zwei Gruppen aufteilen: analytische und numerische Methoden [85]. Für analytische Verfahren existiert eine exakte Lösung, wohingegen numerische Verfahren immer nur eine Annäherung an die Lösung erlauben. Ihr Vorteil besteht in der einfachen Implementierung in Computerprogrammen. Zusätzlich steigt der Aufwand für eine analytische Lösung mit zunehmender Systemkomplexität stark an. Im Gegensatz dazu eignen sich numerische Methoden zum Lösen komplexer Systeme, wobei komplexe Systeme sich durch multiple Zustände, Nichtlinearitäten und heterogene physikalische Zusammenhänge auszeichnen.

Bedingt durch die Leistungssteigerung in der Computertechnik sowie die Eignung von Computern zur numerischen Lösung komplexer Systeme, hat sich der Einsatz von Simulationstechniken in nahezu allen technischen Bereichen etabliert. Die wichtigsten Argumente für den Einsatz der Simulationstechnologie und ihre Hauptvorteile liegen in der Steigerung der Entscheidungsqualität und -sicherheit. Gleichzeitig hilft die Simulation dabei, kritische Entscheidungen in früheren Entwicklungsphasen treffen zu können. Daher ist aus wirtschaftlicher Sicht vor allem bei komplexen Systemen die Simulation in allen Entwicklungsphasen anzuraten. Je komplexer ein System ist, umso mehr rechtfertigt sich der Einsatz von Simulationen [84]. Insbesondere Flugzeugsysteme zeichnen sich durch eine hohe Komplexität aus. Zudem sind Tests an realen Flugzeugsystemen kostenintensiv und können erst in den späten Phasen der Entwicklung durchgeführt werden. Um die vollen Potenziale des Simulationseinsatzes zu nutzen und in der Flugzeugentwicklung sinnvoll einsetzen zu können, sind zunächst die Vor- und Nachteile zu betrachten, um entsprechende Voraussetzungen für den Simulationseinsatz schaffen zu können (siehe Tabelle 2-1).

Es wird deutlich, dass der Simulationseinsatz neben einer Reihe von Vorteilen auch gewisse Risiken birgt. Grundsätzlich sollte eine Simulationsstudie nur mit entsprechenden Fachkenntnissen nach sorgfältiger Planung des Vorgehens, der Analyse des Zustands und zum richtigen Zeitpunkt durchgeführt werden. Zusätzlich ist die Qualität der Simulationsergebnisse zum einen stark von der Qualität des Datenmaterials und zum anderen wesentlich von der Korrektheit des Simulationsmodells abhängig [84]. An dieser Stelle sei darauf hingewiesen, dass in allen Fällen darauf zu achten ist, dass ein Simulationsmodell immer nur eine Nachbildung des realen Systems darstellt und somit zu fehlerhaften Ergebnissen führen kann (lat.: simulare = vortäuschen) [21].

2.1 Grundlagen der Simulationstechnik

Vorteile	Nachteile
<ul style="list-style-type: none"> + Erhöhung der Entscheidungsqualität und Entscheidungssicherheit + Reduzierung der Investitionskosten + Ungefährliche Untersuchung von kritischen Systemzuständen + Schnelle Parametervariation (flexibel) + Einfache und überschaubare Systemanalyse mit geringem finanziellen Aufwand + Verbesserung des Systemverständnisses 	<ul style="list-style-type: none"> – Intensiver Modellierungsaufwand – Zusätzliche Validierung der Simulationsmodelle nötig – Gewährleistung einer optimalen Lösung nicht möglich

Tabelle 2-1 Vor- und Nachteile der Simulationstechnik

Zwar sind dem Einsatz von Simulation Grenzen gesetzt, allerdings überwiegen die in Tabelle 2-1 enthaltenen Vorteile deutlich und rechtfertigen den Einsatz von Simulation in der Systementwicklung. Im folgenden Abschnitt werden zunächst die Grundlagen zum Thema Simulation vorgestellt.

Simulationen lassen sich prinzipiell in zwei Gruppen teilen (siehe Abbildung 2-1) die analoge Simulation und die digitale Simulation. Die analoge Simulation zeichnet sich dadurch aus, dass alle Werte innerhalb des Wertebereichs des gewählten Simulationsmodells zu jedem Zeitpunkt zur Verfügung stehen (z. B. Analogrechner mit Operationsverstärkern).

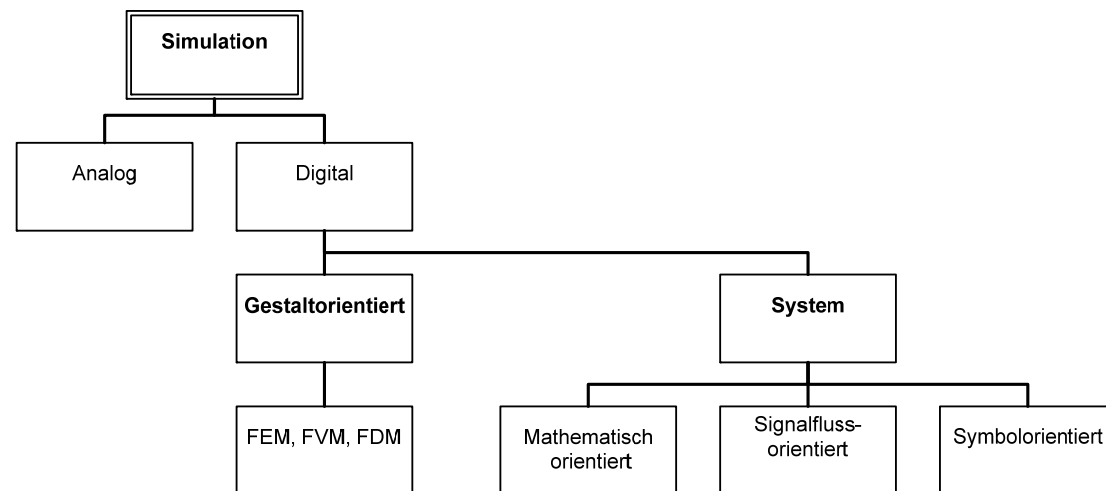


Abbildung 2-1 Simulationsorganigramm

In der digitalen Simulation werden die Werte nur zu diskreten Zeitpunkten berechnet. Die Schrittweite der Simulation bestimmt den nächsten Berechnungszeitpunkt, für eine unendlich kleine Schrittweite erhält man wiederum eine analoge Simulation.

$$\Delta t = \lim_{x \rightarrow 0} x \quad (2-1)$$

Die digitale Simulation sollte hierbei nicht mit der diskreten Simulation (zeitdiskret) verwechselt werden. Die diskrete Simulation benutzt die Zeit, um nach statischen oder zufällig bemessenen Zeitintervallen bestimmte Ereignisse hervorzurufen, welche ihrerseits den Systemzustand bestimmen. Ihr gegenüber steht die kontinuierliche Simulation, die im Allgemeinen mithilfe von Differenzialgleichungen unter Zuhilfenahme der numerischen Integration den Systemzustand zum jeweiligen Zeitpunkt approximiert (vgl. Kapitel 7.1). Hierfür wird in jedem Zeitschritt eine durch das Modell beschriebene Zustandsänderung des Systems hervorgerufen. Die kontinuierliche Simulation ist deterministisch, d. h., die Simulation gelangt in Abhängigkeit von bestimmten Parametern zu einem Ergebnis. In Abbildung 2-2 sind die kontinuierliche und die diskrete Simulation dargestellt. Es wird deutlich, dass sich der Systemzustand der diskreten Simulation zwischen zwei Ereignissen nicht ändert. Für eine kleine Schrittweite entsteht durch die lineare Approximation zwischen den Stützstellen der Eindruck einer ununterbrochenen Wellenform.

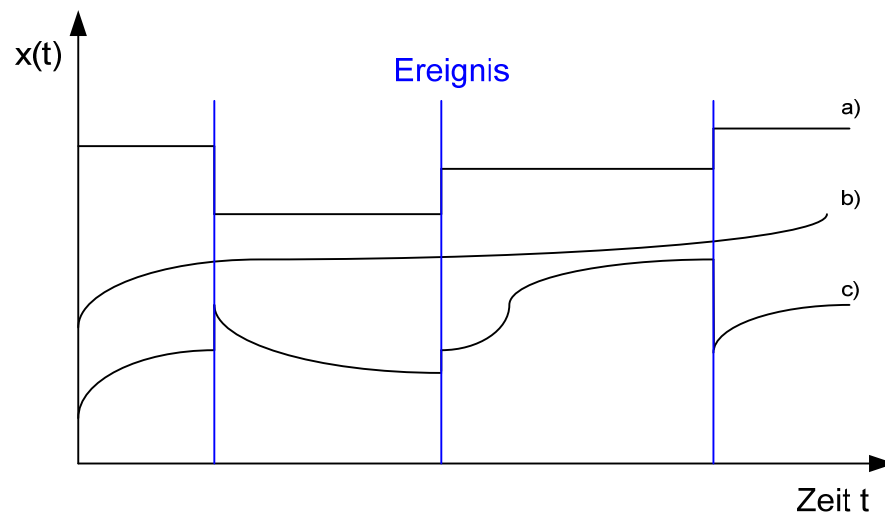


Abbildung 2-2 Beispiele von Signalverläufen a) diskret, b) kontinuierlich, c) hybrid

Von einer hybriden Simulation spricht man, wenn das Modell sowohl kontinuierliche als auch zeitdiskrete Zustandsänderungen aufweist. Da im Zeitpunkt des Ereignisses die Differenzierbarkeit nicht gegeben ist, darf nicht über ein Ereignis hinweg integriert werden. Der Ablauf einer hybriden Simulation lässt sich in die folgenden vier Phasen einteilen [85]:

1. Integration der Differenzialgleichungen
2. Detektieren eines Ergebnisses
3. Propagieren des Ergebnisses
4. Identifizieren konsistenter Anfangsbedingungen

Die kontinuierliche Simulation und damit die Integration von Differenzialgleichungen wird so lange ausgeführt, bis ein diskretes Ereignis detektiert wird. Vor der Wiederaufnahme der Integration muss sichergestellt sein, dass alle anliegenden Ereignisse abgearbeitet sind und konsistente Anfangsbedingungen für die Integration vorliegen. Diese vier Phasen werden so lange durchlaufen, bis das Ende der Simulation erreicht wird. Bei der hybriden Simulation wird also die Kontrolle über das Voranschreiten der Simulation zwischen der kontinuierlichen und der diskreten Simulation hin und her gegeben. Für zusätzliche Informationen zum Thema Behandlung von Zustandsereignissen sei an dieser Stelle auf die Quelle [85] verwiesen.

Die computergestützte (digitale) Simulation lässt sich in zwei weitere Klassen einteilen (siehe Abbildung 2-1): die gestaltorientierte und die Systemsimulation [183]. Die gestaltorientierte Simulation basiert zum Großteil auf Methoden der Finiten-Elemente (FEM) oder der Finiten-Volumen (FVM). Für diese Simulation wird immer eine Systemgeometrie, also eine Gestalt des Systems, benötigt. Im Unterschied hierzu wird mithilfe der Systemsimulation die Funktionalität des Systems analysiert. Aus dieser Aufteilung wird ein Dilemma offensichtlich. Im Entwicklungsprozess laufen diese beiden Simulationsmethoden überwiegend getrennt voneinander ab, dieses Problem wird im Kapitel 3 genauer untersucht [183]. Eine virtuelle Analyse der Interaktionen zwischen der Funktionalität und der Gestalt des Systems erfordert die Kopplung dieser Simulationsklassen. Aus diesem Grund wird im Kapitel 5.2.3 eine lose Kopplung zwischen der gestaltorientierten und der Systemsimulation beschrieben.

Die Systemsimulation lässt sich nach Brychat wiederum in mathematisch orientierte, signalflussorientierte und symbolorientierte Simulation unterteilen (vgl. Abbildung 2-1) [21]. Die Simulationsprogramme dieser drei Klassen unterscheiden sich in der Darstellungsform und Modellierung des zu simulierenden Systems. In mathematisch orientierten Simulationsprogrammen wird das System in mathematischer Form modelliert. Eine grafische Darstellung des Systems, z. B. in Form eines Blockschaltbilds, ist nicht vorhanden. Programme die zu dieser Klasse von Simulationsprogrammen gehören, sind Matlab, Mathematica, Mathcad und Maple [4], [163]. Eine häufig verwendete Beschreibungsform der signalflussorientierten Programme ist die Darstellung mithilfe von Blockschaltbildern [11]. Hierfür wird das Verhalten des Systems in einer grafischen Darstellung erfasst, die aus den einzelnen Teilblöcken sowie den Signalflüssen zwischen den Blöcken besteht. Aus der Blockübertragungsfunktion und der Eingangsgröße wird die Ausgangsgröße berechnet [152]. Die Blöcke werden in der

2.1 Grundlagen der Simulationstechnik

Simulation also sequenziell in einer festen Reihenfolge berechnet. Diese Simulationsprogramme eignen sich für die Analyse regelungstechnischer Anwendungen. Die symbolorientierten Simulationsprogramme stellen eine grafische Modellierung des physikalischen Systems bereit. In anschaulicher Form entspricht das grafische Modell von der Symbolik her weitestgehend dem physikalischen Modell [26]. Die Verbindungen zwischen den Komponenten stellen den physikalischen Zusammenhang her. Der Vorteil der symbolorientierten Simulationsprogramme besteht in der akusalen Modellierung [57]. Die Manipulation der Systemgleichungen erfolgt automatisch. Im Vergleich dazu müssen in den mathematischorientierten und signalfluss- Simulationsprogrammen die Systemgleichungen manuell aufgestellt und aufgelöst werden. Im Anhang sind eine detailliert Beschreibung der verschiedenen Simulationsklassen sowie die mathematischen Grundlagen der Simulationstechnik enthalten (vgl. Kapitel 7). Zudem werden anhand eines Feder-Masse-Systems die unterschiedlichen Modellierungsweisen der drei Systemsimulationsprogramme verdeutlicht.

2.2 Systementwicklungsprozess

Das Ziel dieser Arbeit ist die Bereitstellung eines Konzepts zur durchgängigen Unterstützung des Systementwicklungsprozesses im Flugzeugbau. Aus diesem Grund werden im Folgenden zunächst bestehende Entwicklungsprozesse und Prozessmodelle beschrieben und deren Einsatz in der Luftfahrtindustrie analysiert. Es folgt eine Untersuchung des Simulationseinsatzes in verschiedenen Industriezweigen. Abgeschlossen wird das Kapitel mit dem Stand der Technik der Informationsbereitstellung.

2.2.1 Allgemeine Prozessmodelle

Es existiert eine Reihe von branchen- und produktunabhängigen Vorgehensmodellen für die Entwicklung von Systemen und Produkten. In der VDI-Richtlinien 2221 wird ein solches Vorgehensmodell zur Verfügung gestellt (Abbildung 2-3) [178]. Durch ihren Einsatz sollen Entwicklungsziele wie die Reduktion des Aufwands, die optimale Einbindung von Datenverarbeitungsmöglichkeiten, die Gewährleistung der Marktfähigkeit der Produkte usw. realisiert werden.

Im ersten Schritt werden Anforderungen an das Produkt bzw. System definiert. Anschließend werden die wichtigsten Funktionen ermittelt, wobei für komplexe Systeme eine Verknüpfung der einzelnen Teilfunktionen zu Funktionsstrukturen, für die im dritten Schritt Lösungsprinzipien zu erarbeiten sind, anzuraten ist. Anschließend werden die Lösungsprinzipien in realisierbare Module gegliedert. In der Konkretisierung (Schritt fünf) werden die Module und im folgenden Schritt das gesamte Produkt gestaltet. Das Ergebnis ist der Gesamtentwurf, der alle wesentlichen Produkteigenschaften enthält. Im letzten Schritt wird die weitere Ausarbeitung geplant. In dieser Methodik zur Entwicklung und Konstruktion technischer Systeme und Produkte ist nicht erkennbar, inwiefern die Besonderheiten komplexer Systeme berücksichtigt werden. Aus diesem Grund wird im Folgenden das Vorgehensmodell näher betrachtet. Nach Pahl lässt sich der Entwicklungsprozess in vier Hauptphasen unterteilen [124]:

1. Planung
2. Konzeption
3. Entwurf
4. Umsetzung

Die Einteilung in Phasen eignet sich zur Handhabung komplexer Prozesse und wird daher auch im Rahmen dieser Arbeit durchgeführt.

Es wird darauf hingewiesen, dass besonders in der Konzeptphase wesentliche Grundlagen für die weitere Entwicklung gelegt werden und grundlegende Mängel am Lösungsprinzip in der Phase der Ausarbeitung nur noch mit großem Aufwand behoben werden können. Bei komplexen Problemstellungen ist anzuraten, das Gesamtsystem in Systemteile und -elemente zu unterteilen und diese anschließend zu einer Gesamtlösung zu verknüpfen [178].

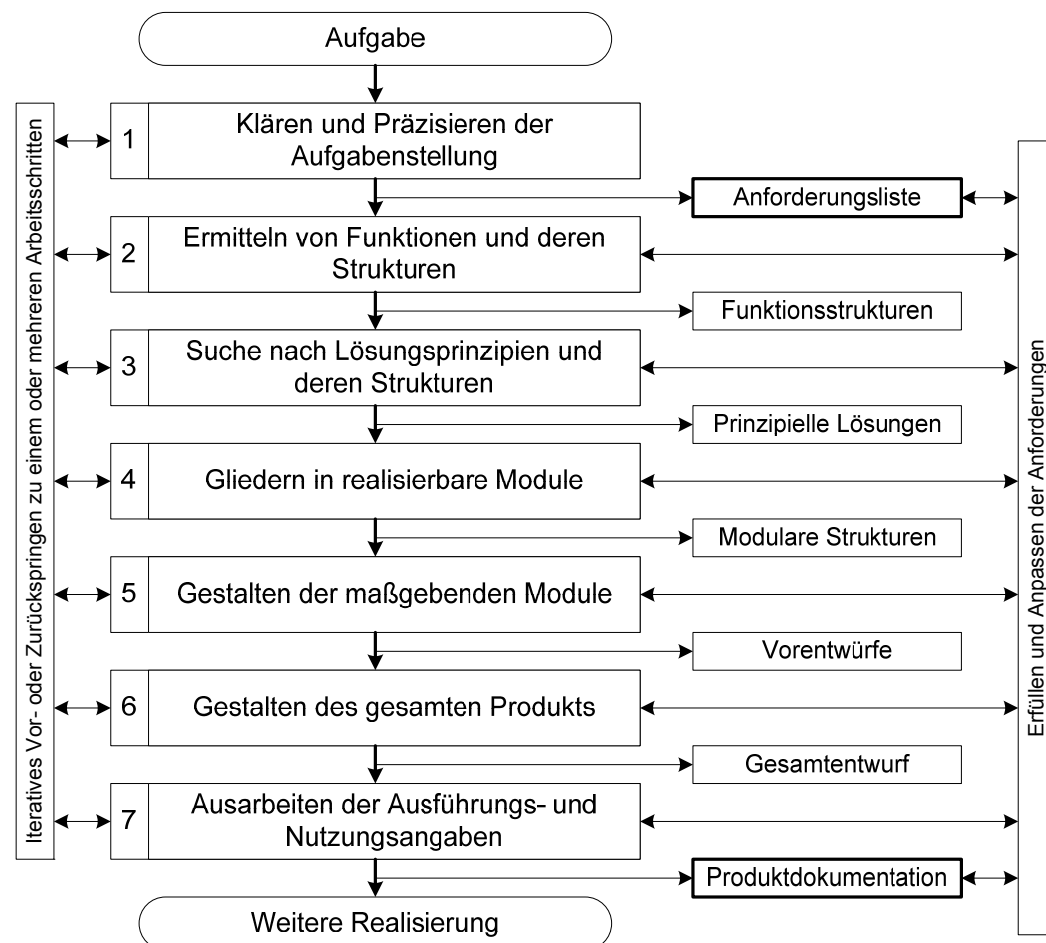


Abbildung 2-3 Generelles Vorgehen beim Entwickeln und Konstruieren nach VDI 2221 [178]

Die Subsysteme bzw. Systemelemente durchlaufen unabhängig voneinander in jeder Entwicklungsphase den sogenannten systemtechnischen Problemlösezyklus (Mikroebene), der Bestandteil des generellen Vorgehensmodells (siehe Abbildung 2-3) ist (Makroebene). Durch den iterativen Charakter dieses Vorgehens werden eine schrittweise Optimierung sowie eine Erhöhung des Informationsniveaus erreicht. Gleichzeitig ist allerdings zu berücksichtigen, dass Iterationen trotz der genannten Vorteile auf ein erforderliches Maß reduziert werden müssen, um den Entwicklungsprozess möglichst effektiv zu durchlaufen. für Informationen zum Thema Problemlösezyklus siehe Quelle [186], [178].

Insgesamt stellt die VDI-Richtlinie 2221 eine generelle Vorgehensweise bei der Entwicklung komplexer Systeme zur Verfügung. Es wird allerdings nicht deutlich, wie der Entwicklungsprozess von hochintegrierten komplexen Systemen, wie sie im Flugzeugbau anzutreffen sind, im konkreten Fall umzusetzen ist. Daher ist im Rahmen dieser Arbeit nach Ansätzen zu suchen, die insbesondere auf die Entwicklung von stark vernetzten Systemen eingehen.

Aus diesem Grund wird im Folgenden das V-Modell XT betrachtet, welches eine standardisierte Methode während des gesamten Produktlebenszyklus für komplexe Produkte zur Verfügung stellt. Der Entwicklungsprozess im V-Modell besteht aus zwei Hauptphasen (vgl. Abbildung 2-4): zum einen aus der Spezifikation und Dekomposition und zum anderen aus der Realisierung und Integration des Systems [23].

Im Rahmen der Spezifizierung des Produkts bzw. Systems werden Anforderungen der vorangegangenen Entwicklungsphasen übernommen und verfeinert. Anschließend werden die Systemelemente spezifiziert und die definierten Anforderungen der nächsten Systemelementebene zugeordnet. Das Produkt wird also vom Groben zum Feinen hin entwickelt (engl.: Top-down-Vorgehen), das bedeutet, dass auch in diesem Vorgehensmodell das System in Subsysteme unterteilt wird, die getrennt voneinander weiterentwickelt werden.

In der Realisierung und Integration des Systems werden die Entwicklungsschritte in umgekehrter Reihenfolge durchlaufen (engl.: Bottom-up-Vorgehen), das heißt, die definierten Systemelemente werden realisiert und zu einem Gesamtsystem integriert.

2.2 Systementwicklungsprozess

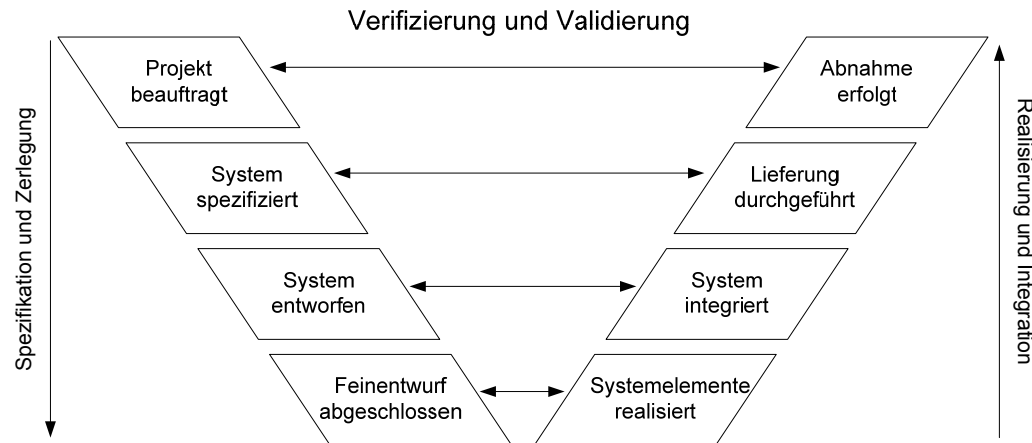


Abbildung 2-4 Entwicklung nach dem V-Modell XT [23]

Ziele der standardisierten Vorgehensweise sind die Minimierung von Projektrisiken, die Verbesserung der Qualität, die Reduktion der Entwicklungskosten und die Verbesserung der Kommunikation [23].

Im Vergleich mit der VDI-Richtlinie 2221 wird ein eklatanter Unterschied deutlich. Das V-Modell XT fordert einen Abgleich der realen Systemelemente mit den entsprechenden Spezifizierungen der Systemelemente auf jeder Detaillierungsstufe, die sogenannte Validierung und Verifikation (V&V). Allerdings wird das Zusammenwirken aller Teilelemente bzw. Subsysteme erst im letzten Schritt der Realisierung überprüft. Inkompatibilitäten zwischen den Teilsystemen werden somit erst spät erkannt, sodass zeitaufwendige und damit teure Rücksprünge erforderlich werden [146], [41], [143], [179], [23].

Zusammenfassend lässt sich sagen, dass das V-Modell XT zwar gute Ansätze hinsichtlich der Validierung und Verifikation im Prozess bietet, für die Entwicklung von hochintegrierten Systemen jedoch ungeeignet ist. Die VDI-Richtlinie 2206 wurde speziell für interdisziplinäre Produktentwicklungen entwickelt, im Folgenden soll überprüft werden, ob sich diese Methodik für den Einsatz in der Luftfahrtindustrie eignet.

Das Vorgehensmodell der VDI-Richtlinie 2206 wurde für Entwicklungsvorhaben im Bereich der Mechatronik entwickelt [179]. Bedingt durch die Beteiligung verschiedener Fachbereiche im Produktentstehungsprozess steigen die Interoperabilität und damit die Komplexität in der Entwicklung [147]. Damit spielt die Definition der Schnittstellen im Produktentstehungsprozess eine wesentliche Rolle. Als Anforderungen an das Vorgehensmodell nach VDI 2206 ergeben sich unter anderem ein frühzeitiger Austausch und Abgleich von Ergebnissen und Modellen sowie die Möglichkeit des Zugriffs auf den aktuellen Daten- und Wissensbestand. Das Vorgehensmodell fasst daher die bisher vorgestellten Vorgehensmodelle zusammen und erweitert diese:

- Allgemeiner Problemlösezyklus (vgl. [178]) auf der Mikroebene: Unterstützung durch flexible und an die Eigenheiten der Entwicklungsaufgabe angepasste Vorgehenszyklen
- V-Modell auf der Makroebene: Vorgehen nach dem V-Modell, das an die interdisziplinären Anforderungen angepasst wurde
- Vordefinierte Prozessbausteine für sich wiederholende Tätigkeiten

Die Besonderheiten des in der VDI-Richtlinie 2206 vorgestellten Vorgehens bestehen in einem mehrmaligen Durchlaufen der Vorgehenszyklen (Iteration), insbesondere der Makrozyklen, bei komplexen Produkten. Zudem wird in der VDI-Richtlinie 2206 betont, dass eine interdisziplinäre Produktentwicklung eine frühzeitige Analyse von Wechselwirkungen erfordert. Ausgehend von der Gesamtstruktur werden die Elemente bzw. Teilsysteme aufgebaut und detailliert und es kommt im Laufe der Detaillierung zu Änderungen an der Gesamtstruktur. Daher sollten iterativ zunächst Kenntnisse über die Gesamtstruktur gewonnen werden, um darauf aufbauend die Elemente bzw. Teilsysteme zu detaillieren (engl.: Top-down-Design). Im Laufe der Detaillierungsphase können wiederum Änderungen an der Gesamtstruktur erforderlich sein, daher wird ein Wechsel von Top-down- und Bottom-up-Design empfohlen. In der VDI-Richtlinie 2206 wird betont, dass die Aufteilung des Gesamt-

systems in Teilsysteme, die getrennt voneinander entwickelt und anschließend wieder zusammengefügt werden (Bottom-up-Design), aufgrund der Wechselwirkungen zwischen den Systemen nicht ausreicht, um die Funktionalität des Gesamtsystems zu garantieren [179].

Das Vorgehensmodell nach VDI 2206 ist damit eine Kombination aus V-Modell auf der Makroebene und Problemlösezyklus (vgl. Seite 7) auf der Mikroebene. Es greift den Grundgedanken des V-Modells, eine Verifizierung und Validierung der Entwicklungsergebnisse auf jeder Detaillierungsebene, auf. Gleichzeitig werden durch das iterative Vorgehen die Besonderheiten von stark vernetzten, domänenübergreifenden Systemen berücksichtigt. Allerdings findet die Verifizierung der Ergebnisse auch im Vorgehensmodell der VDI-Richtlinie 2206 erst im Rahmen der Realisierung statt. Ebenso fehlt ein konkreter Vorschlag zur Umsetzung der Vernetzung der verschiedenen Systeme in der Entwicklung.

Identifikation von Defiziten herkömmlicher Prozessmodelle

Basierend auf den im vorangegangenen Abschnitt vorgestellten Prozessmodellen, weisen herkömmliche Entwicklungsprozesse eine zu späte Verifikation auf, die zu zeit- und kostenintensiven Änderungsschleifen führt. Eine weitere Schwachstelle ist die zu kurze Konzeptphase bzw. eine zu frühe Festlegung auf ein Konzept. Die frühen Entwicklungsphasen beeinflussen die Gesamtentwicklung hinsichtlich der Zeit und der Kosten maßgeblich. Es ist daher von entscheidender Bedeutung, bereits in diesem frühen Entwicklungsstadium das am besten geeignete Konzept zu ermitteln [41], [175]. Wird sich zu früh auf ein Konzept festgelegt, ohne weitere Alternativen in Betracht zu ziehen, können in der folgenden Detaillierung grundsätzliche Mängel des Konzepts kaum ausgeglichen werden. In der Literatur wird aus diesem Grund die Bedeutung einer fundierten Konzeptentwicklung betont [124].

Runge fordert den Einsatz von softwarebasierten Entwurfstechniken in der Systemdefinition, um den Entwicklungsaufwand in frühe Projektphasen zu verlagern und eine optimale Konzeptauswahl zu gewährleisten [143]. An dieser Stelle sei darauf hingewiesen, dass sich der Einsatz von Modellen bzw. Simulation in der frühen Entwicklungsphase, bedingt durch die geringe Systemkenntnis und entsprechend wenigen Daten, als kompliziert erweist. Aus diesem Grund würden eine Standardisierung von Schnittstellen und Austauschformaten, eine Kopplung von Werkzeugen und eine Codeintegration die Datenverfügbarkeit unterstützen sowie eine Entwicklung über bestehende Grenzen der Entwurfsbereiche hinaus ermöglichen [41], [124], [192].

Bei der Entwicklung komplexer multidisziplinärer Systeme bzw. Produkte sind häufige und kleine Verifikations- und Validierungsintervalle anzuraten, um zeit- und kostenintensive Iterationsschleifen zu vermeiden. In Abbildung 2-5 sind diese Iterationsintervalle nach Schäfer dargestellt. Besonders in der Spezifikationsphase kann mit virtuellen Prototypen oder Softwareprogrammen die Interaktionen der Teilsysteme bzw. Komponenten frühzeitig analysiert und korrigiert werden [146].

Der Ansatz von Schäfer sowie das Vorgehensmodell der VDI 2206 machen deutlich, dass eine auf die Besonderheiten von hochintegrierten Systemen abgestimmte Vorgehensweise erforderlich ist. Es ist dabei von essenzieller Bedeutung, die Entwicklungstätigkeiten in der frühen Entwicklungsphase zu intensivieren, um so teure Änderungsschleifen zu vermeiden. Es bestehen zwar Lösungsansätze, wie die von Schäfer vorgeschlagene frühe Verifizierung der Entwicklungsergebnisse durch virtuelle Modelle, allerdings fehlt bislang die konkrete Umsetzung. Das folgende Kapitel enthält einen Überblick der Systementwicklung in der Flugzeugindustrie.

2.2 Systementwicklungsprozess

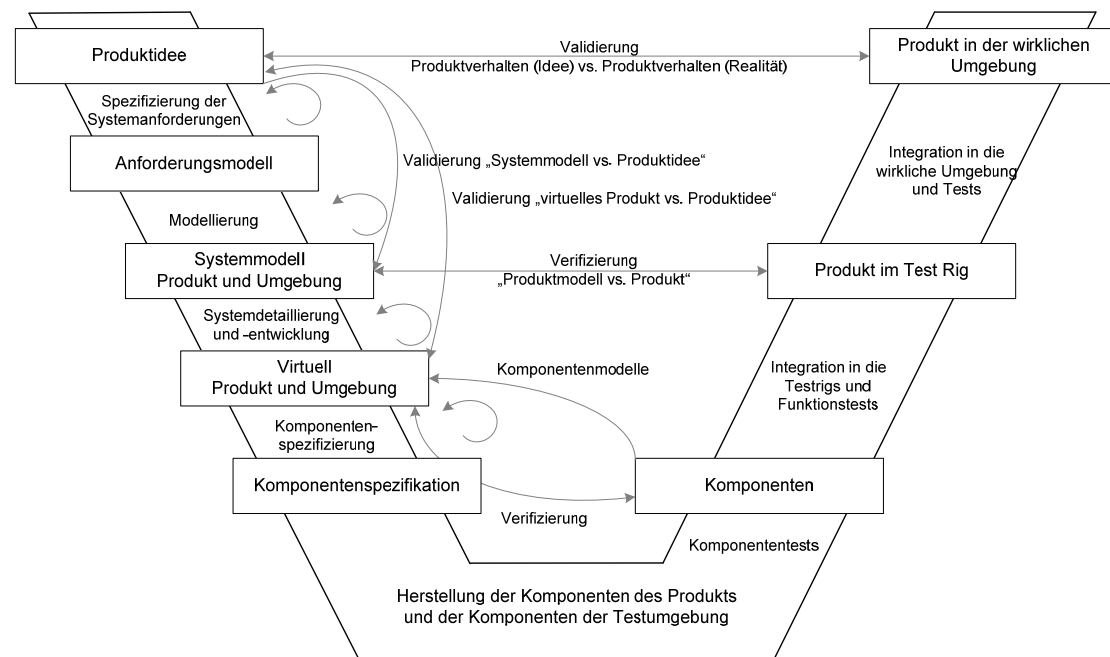


Abbildung 2-5 Validierung und Verifizierung (V&V) im V-Modell nach Schäfer [146]

2.2.2 Flugzeugsystementwicklungsprozess

Nachdem zuvor allgemeine Vorgehensmodelle betrachtet wurden, wird in diesem Kapitel der Entwicklungsprozess im Flugzeugbau dargestellt.

Die Flugzeugentwicklung zeichnet sich durch hohe Ansprüche an die Qualität sowie die Komplexität des Systems aus. Um ein konkurrenzfähiges Produkt anbieten zu können, besteht der Anspruch, allen Kundenwünschen nachzukommen und gleichzeitig eine stabile und sichere Leistung der Systeme zu gewährleisten. Die hohe Projektkomplexität resultiert aus der großen Anzahl der an der Flugzeugentwicklung beteiligten Personen, funktionalen Bereiche und beteiligten Zulieferer sowie deren Interdependenzen [161].

Im Vordergrund der Entwicklung steht die Entwicklungsqualität und damit häufig die Verwendung von bekannten, bereits getesteten Technologien. Technische Risiken sind dadurch kalkulierbar, auf den Innovationsgrad der Entwicklung wirkt sich das allerdings negativ aus. Die Innovationsfähigkeit ist für die Flugzeugentwicklung von untergeordneter Bedeutung, sollte aber aus Gründen der Konkurrenzfähigkeit dennoch nicht unberücksichtigt bleiben. Es ist daher zu untersuchen, wie die Entwicklungsqualität erhalten werden kann, ohne ausschließlich auf existierende Technologien zurückgreifen zu müssen.

Für die Flugzeugentwicklung wurde von der Engineering Society for Advancing Mobility Land Sea Air and Space (Abk. SAE) die Richtlinie ARP 4754 (engl.: Aerospace Recommended Practice) entwickelt. Das Ziel dieser internationalen Standardrichtlinie ist es, die Erfüllung der Flugtauglichkeit zu gewährleisten. Ursprünglich wurde sie zur Entwicklung von elektronischen Systemen konzipiert, sie lässt sich aber ohne Weiteres für sämtliche hochintegrierten und komplexen Systeme im Flugzeug einsetzen [156].

Wie bei den bereits vorgestellten Vorgehensmodellen in Kapitel 2.2.1 beschreibt die ARP 4754 eine iterative und keine sequenzielle Vorgehensweise. Zunächst wird aus den Flugzeuganforderungen das funktionale System, das heißt die Umsetzung, abgeleitet. Der Systementwicklungsprozess nach ARP 4754 besteht aus folgenden Aktivitäten [156]:

1. Identifizieren der funktionalen Anforderungen und Schnittstellen auf Flugzeugebene
2. Bestimmung der Konsequenzen und Auswirkungen bei Ausfall von Funktionen
3. Zuordnen von Funktionen zu Systemen und Personen
4. Design der Systemarchitektur und Zuordnung von Anforderungen zu Punkten
5. Zuordnung von Anforderungen zu Hardware (Abk. HW) bzw. Software (Abk. SW)
6. Design und Erstellen der HW bzw. SW
7. HW- bzw. SW-Integration
8. Systemintegration

Ein Vorteil der ARP 4754 liegt in der Berücksichtigung der Sicherheitsanforderungen von Flugzeugsystemen und damit in der Betonung der Entwicklungsqualität. In Analogie zu den allgemeinen Vorgehensmodellen wird für die Detaillierung ebenfalls ein Top-down-Vorgehen eingesetzt, um die Teilsysteme im Rahmen der Umsetzung zu integrieren. Über die konkrete Umsetzung, wie die tatsächliche Vernetzung der unterschiedlichen Flugzeugsysteme in der Entwicklung erreicht werden kann, werden allerdings keine Aussagen gemacht.

Die ARP 4754 bildet die Grundlage für den Entwicklungsprozess im Flugzeugbau. Die einzelnen Flugzeughersteller adaptieren diese Vorgehensweise für ihre spezifischen Bedürfnisse. Die jeweiligen Entwicklungsprozesse lassen sich ebenfalls im V-Modell darstellen (vgl. Abbildung 2-4), dieses wird durch die kontinuierliche Validierung und Verifizierung (V&V) vom Konzept bis zum fertigen Produkt betont.

Durch die späte Überprüfung der Teilsysteme im Rahmen der Realisierung kann es, bedingt durch Entwicklungsfehler, zu zeit- und kostenintensiven Änderungsschleifen kommen. Zusammenfassend lässt sich sagen, dass die allgemeinen Vorgehensmodelle und die Entwicklungsprozesse im Flugzeugbau den Ablauf des prinzipiellen Systementwicklungsprozesses darstellen sowie die Besonderheiten bei der Entwicklung von hochintegrierten Systemen in Ansätzen berücksichtigen. Allerdings fehlen konkrete Umsetzungsvorschläge, wie die Vernetzung zwischen integrierten Teilsystemen erreicht werden kann. Daher erweist sich die Entwicklung eines systemübergreifenden Prozessmodells als erforderlich.

2.3 Einsatz von Simulation im Entwicklungsprozess

Klassische Entwicklungsprozesse zeichnen sich dadurch aus, dass erst in den Phasen der Detaillierung und Umsetzung Wissen über das Systemverhalten erlangt wird. In frühen Phasen basiert die Entwicklung auf empirischem Wissen. Für Neuentwicklungen, die sich dadurch kennzeichnen, dass kein erfahrungsbasiertes Wissen vorhanden ist und diese sich nicht von ähnlichen Produkten ableiten lassen, ist die Entwicklung aus diesem Grund mit Risiken verbunden. Im klassischen Entwicklungsprozess wird der Einsatz von Simulation kaum berücksichtigt [2], [107].

In Abbildung 2-6 sind der Einfluss auf die Entwicklungskosten sowie die Entwicklungskosten in Abhängigkeit der Entwicklungszeit dargestellt. Es wird deutlich, dass bis zu 70 % der Produktentwicklungskosten bereits in den frühen Phasen der Entwicklung festgelegt werden. Demgegenüber fallen in diesen Phasen nur 5 % der Entwicklungskosten an. Aus diesem Grund sollte der Schwerpunkt des Simulationseinsatzes in diesen frühen Phasen liegen [84]. Zunächst resultiert aus dem Einsatz von Simulation ein Mehraufwand, der mittelfristig durch eine Reduktion der Entwicklungszeit und -kosten amortisiert wird. Durch den Einsatz der Simulation erhöht sich zudem die Kenntnis über das Systemverhalten und trägt damit zur Erhöhung der Produktqualität bei [2], [42], [107]. In Tabelle 2-2 ist ein Überblick der Aufgaben und der Nutzen des Simulationseinsatzes in den verschiedenen Entwicklungsphasen dargestellt.

In der Konzeptentwicklung sollte die Simulation zum Vergleich von Konzeptvarianten eingesetzt werden, um qualitative Prognosen abzugeben und konzeptionelle Schwachstellen an den Konzeptideen zu erkennen. Zudem erhöht eine rechnergestützte Konzeptüberprüfung das Wissen über das Systemverhalten. Besonders bei der Entwicklung von multidisziplinären Produkten oder Systemen ist eine Überprüfung der Kompatibilität mithilfe der Simulation sinnvoll [2], [42], [73], [54], [139].

In der Detaillierungsphase unterstützt der Simulationseinsatz die Ausgestaltung des Konzepts, die Überprüfung wichtiger Anforderungen sowie die Absicherung von Funktionen. Eine virtuelle Verifikation der entwickelten Elemente sollte die Entwicklungsarbeit bestätigen [107].

In der anschließenden Komponentenentwicklung können an die reale Komponente angepasste Modelle für sichere Prognosen über das Verhalten des realen Systems eingesetzt werden [107].

In der Test- und Integrationsphase wird Simulation in den Hardware in the Loop-Tests (Abk. HiL) eingesetzt. Dabei wird unter anderem die reale Hardware in einer Simulationsumgebung getestet. Dies ist erforderlich, wenn das reale System aus Sicherheits- oder Kostengründen nicht untersucht werden kann. Zur Abbildung fehlender realer Komponenten können Simulationsmodelle aus der Komponentenentwicklung bzw. dem Prototypenbau in die Tests eingebunden werden [2], [36], [42], [139].

2.3 Einsatz von Simulation im Entwicklungsprozess

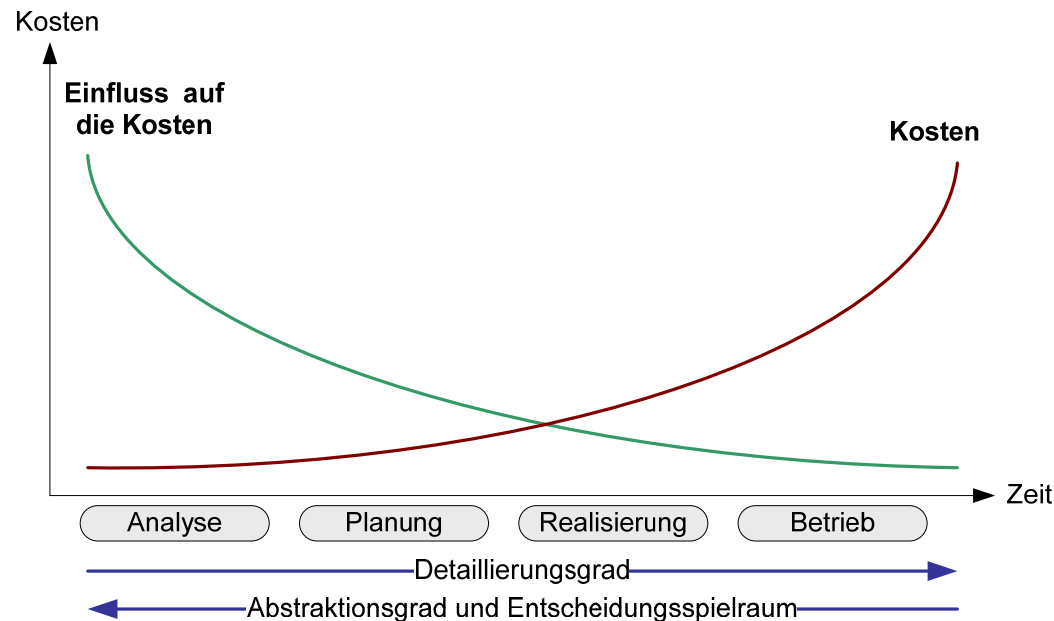


Abbildung 2-6 Möglichkeit der Einflussnahme auf die Entwicklungskosten in Abhängigkeit von der Entwicklungsphase [84]

PHASEN	AUFGABEN	NUTZEN
Konzeptentwicklung	<ul style="list-style-type: none"> • Qualitative Prognosen • Vergleich von Konzeptvarianten • Erkennen von konzeptionellen Schwachstellen 	<ul style="list-style-type: none"> • Unterstützen des Erfahrungswissens durch quantifizierbare Berechnungsergebnisse • Abstimmung des Konzepts
Detaillierung (Planung)	<ul style="list-style-type: none"> • Detailanalysen des Systemverhaltens 	<ul style="list-style-type: none"> • Überprüfen von Anforderungen • (Virtuelle) Verifizierung
Komponenten- und Prototypenentwicklung	<ul style="list-style-type: none"> • Fehleranalysen • Detailanalyse 	<ul style="list-style-type: none"> • Prognosesichere Berechnungen des Systemverhaltens
Tests und Integration der Systeme	<ul style="list-style-type: none"> • Vergleich der realen Eigenschaften mit den prognostizierten Eigenschaften • Auffinden von Schwachstellen 	<ul style="list-style-type: none"> • Abgleichen und Verbessern vorangegangener Simulationen mit Messergebnissen • Klärung bei Problemen • Wenn Untersuchungen am realen System nicht möglich sind

Tabelle 2-2 Aufgaben des Simulationseinsatzes in der Entwicklung [42], [107]

Ein Verzicht auf Experimente und Tests am realen System wird durch die Simulation nicht erreicht, zusätzlich sind verschiedene Tests behördlich vorgeschrieben. An dieser Stelle sei darauf hingewiesen, dass die Ergebnisse der Simulation auf Richtigkeit zu prüfen sind und die Modelle mithilfe von Messungen und Experimenten validiert werden müssen. Daher werden Experimente am realen System durch Simulation ergänzt, um die Entwicklungskosten zu senken und die Qualität zu erhöhen [2], [36], [42], [139].

Es wird deutlich, dass der kontinuierliche Einsatz von Simulation während des gesamten Entwicklungsprozesses anzuraten ist. Insbesondere ist hier das hohe Verbesserungspotenzial durch die Simulationsunterstützung in den frühen Phasen der Entwicklung zu nennen. In der Literatur werden prinzipielle Verbesserungsvorschläge zum Einsatz von Simulation während der Produktentwicklung gemacht. Eine konkrete Umsetzung des Simulationseinsatzes während der Entwicklung fehlt allerdings.

2.3.1 Modelldetaillierung im Entwicklungsprozess

Der Entwicklungsprozess von Flugzeugsystemen basiert auf dem V-Modell, welches im linken Strang einem Top-down-Design folgt. Ausgehend von den Hauptsystemanforderungen wird das System in immer kleinere Subsysteme bis hin zur Systemkomponente zerlegt. Für die durchgängige Simulationsunterstützung während dieser Systemdekomposition sind somit Modelle für die unterschiedlichen Detaillierungsebenen erforderlich. Aus diesem Grund ist darauf zu achten, dass sich die Modelle analog zum verfügbaren Datenmaterial und damit zur jeweiligen Entwicklungsphase verhalten und somit mit voranschreitender Entwicklung der Abstraktionsgrad der Modelle abnimmt [188], [55].

Entsprechend der Unterteilung des Entwicklungsprozesses in Phasen lassen sich die Simulationsmodelle in Abstraktionsebenen untergliedern. In Abbildung 2-7 sind die vier Modellabstraktionsebenen Architektur-, Funktions-, Verhaltens- und Komponentenebene dargestellt. Diese Untergliederung bietet eine Möglichkeit, Simulationsmodelle den Phasen im Entwicklungsprozess zuzuordnen und damit eine Klassifizierung der Simulationsmodelle durchzuführen, um so den Modellaustausch zwischen verschiedenen Systemen zu unterstützen. Die Abstraktionsebenen können sowohl in einem Modell hinterlegt sein als auch durch mehrere einzelne Modelle repräsentiert werden. Dabei ist darauf zu achten, dass die Modelle der verschiedenen Abstraktionsebenen unabhängig voneinander genutzt werden können [93].

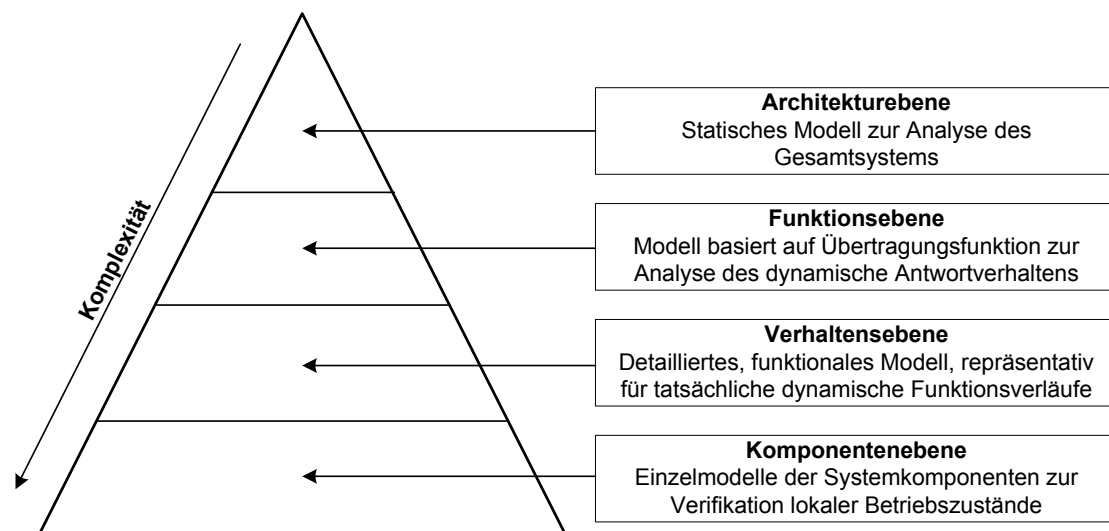


Abbildung 2-7 Modellabstraktionsebenen [93]

Bei den Modellen der Architekturebene handelt es sich um sehr einfache Modelle, deren Erstellung einen geringen Ressourceneinsatz erfordert. Sie eignen sich für den Vergleich von Lösungsalternativen in der frühen Entwicklung und tragen dazu bei, die den Anforderungen am besten entsprechende Lösung für die weitere Entwicklung zu evaluieren. Für den Fall des Einsatzes von Systemsimulatoren handelt es sich bei den Modellen der Architekturebene um statische Modelle, die sich z. B. für die Wirkungsgrad-, Stoffstrom- und Energieberechnung des Gesamtsystems eignen. In der Phase des Systementwurfs (siehe Abbildung 2-4) kommen zunächst funktionale und später während des Feinentwurfs Verhaltensmodelle zum Einsatz; diese Modelle eignen sich z. B. für dynamische Systemuntersuchungen. Die Modelle basieren auf dem stetig wachsenden Datenmaterial, daraus folgt eine Zunahme der Modellkomplexität im voranschreitenden Entwicklungsprozess. In der Phase des Systementwurfs findet bereits eine Systemdekomposition statt. Diese hat zur Folge, dass das System in Subsysteme unterteilt wird und somit auch die Modelle nur einen Teilausschnitt des Gesamtsystems repräsentieren. Aus diesem Grund ist für eine Gesamtsystemanalyse die Kopplung der Subsystemmodelle erforderlich.

Ein System setzt sich aus Subsystemen, die wiederum aus Komponenten bestehen, zusammen. Somit werden die komplexesten Modelle in der Phase der Komponentenentwicklung erstellt (vgl. Abbildung 3-2). Mithilfe dieser Modelle ist eine detaillierte Analyse der Komponenten möglich. Für die Analyse des Gesamtsystemverhaltens ist eine Kopplung sämtlicher Komponentenmodelle erforderlich. Aus diesem Grund wird im folgenden Kapitel die Simulationskopplung multidisziplinärer Systeme beschrieben.

2.3 Einsatz von Simulation im Entwicklungsprozess

2.3.2 Modellierung und Simulation multidisziplinärer Systeme

Wie bereits beschrieben, werden komplexe Systeme in Teilsysteme zerlegt und getrennt voneinander entwickelt (vgl. Kapitel 2.2). Damit erfolgt die Simulation der Teilsysteme zunächst ebenfalls getrennt voneinander. Für die Erhöhung der Entwicklungsqualität sowie für die Reduzierung der Entwicklungszeit ist es allerdings erforderlich, das Verhalten der multidisziplinären Teilsysteme im Gesamtkontext zu betrachten und Wechselwirkungen zwischen den Teilsystemen zu berücksichtigen [118]. Im Folgenden werden verschiedene Varianten der domänenübergreifenden Simulation beschrieben, durch deren Einsatz eine virtuelle Analyse der Wechselwirkungen in einem systemübergreifenden Entwicklungsprozess ermöglicht wird.

Für nahezu alle technischen Problemstellungen existieren heute unzählige spezialisierte Simulationsprogramme, mit deren Hilfe einzelne physikalische Disziplinen modelliert und simuliert werden können. Für jedes zu simulierende Problem wurden bisher einzelne, in sich geschlossene und nicht wiederverwendbare Modelle erstellt [106]. Zudem werden die zu simulierenden Modelle immer komplexer, und es müssen disziplinübergreifende Systemeigenschaften berücksichtigt werden. Diesen Anforderungen kann spezialisierte Simulationssoftware nicht oder nur mit sehr hohem Mehraufwand gerecht werden. Aus diesem Grund gehen immer mehr Simulationssoftwarehersteller dazu über, Schnittstellen zu anderen Simulationsprogrammen zur Verfügung zu stellen, um die Stärken einzelner Programme im Verbund mehrerer Simulatoren nutzen zu können [42].

In den letzten Jahren kam zu der Klasse der spezialisierten Simulationsprogramme eine weitere Klasse, die sogenannten multidisziplinären Simulationsprogramme, hinzu. Sie sind nicht auf einen physikalischen Bereich spezialisiert, sondern erlauben die Modellierung und Simulation von heterogenen physikalischen Systemen [152].

Domänenübergreifende Simulationen lassen sich, abhängig von der Anzahl der erstellten Modelle und verwendeten Simulatoren, prinzipiell in vier Gruppen einteilen, von denen die erste bis dritte Variante näher untersucht werden sollen (vgl. Abbildung 2-8). Die vierte Variante, Modellseparation, wird nicht betrachtet, da sie zur Unterstützung von domänenübergreifenden Entwicklungen nicht geeignet ist, weil das Ziel gerade das Zusammenschalten von Modellen ist. Diese Variante eignet sich z. B. für die logische Aufteilung eines Modells auf ein Mehrrechnersystem. Dieses Vorgehen kommt z. B. in der CFD-Simulation zur Anwendung, um die Rechenzeit zu reduzieren [59].

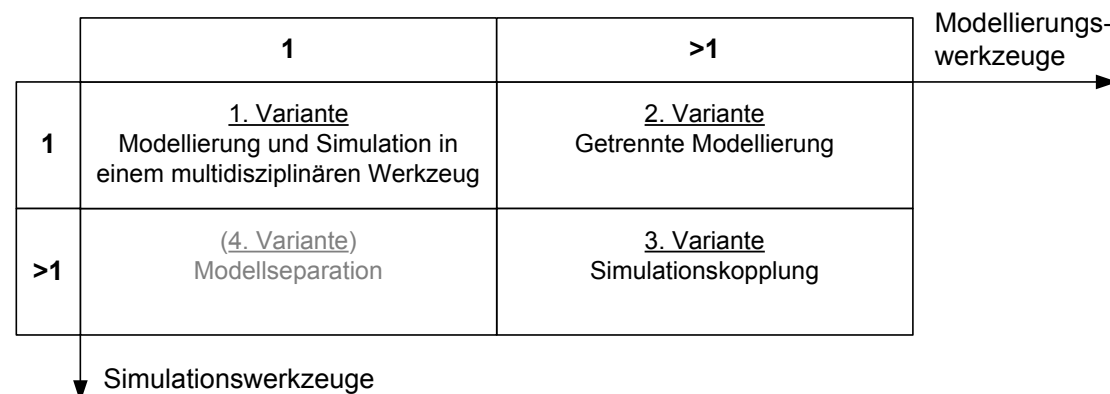


Abbildung 2-8 Varianten der multidisziplinären Simulation [42]

1. Variante: Modellierung und Simulation in einem multidisziplinären Werkzeug

In diesem Fall werden alle Teilsysteme des domänenübergreifenden Systems in einem universellen Programm (z. B. einem multidisziplinären Simulationsprogramm) vollständig modelliert und anschließend simuliert. Der Vorteil dieser Methode besteht darin, dass es zu keinen programmspezifischen Schnittstellenproblemen kommen kann. Wird diese Methode im Entwicklungsprozess eingesetzt, bedeutet dieses zusätzlich, dass die Kommunikation der an der Modellierung und Simulation Beteiligten verbessert wird, da das Modell von allen verstanden wird. Die Verwendung eines einzigen Programms hat aber auch Nachteile. Es kann nicht das für den Anwendungsfall am besten geeignete Programm verwendet werden und zusätzlich wird eine Monopolstellung des jeweiligen Softwareherstellers unterstützt [42], [118].

2. Variante: getrennte Modellierung, aber Gesamtsimulation (Modellimport)

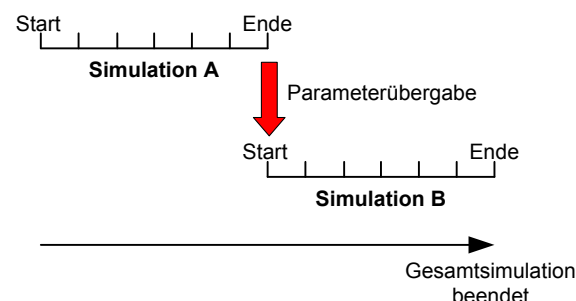
Das Gesamtsystem wird in unterschiedliche Teilsysteme aufgeteilt, die getrennt voneinander modelliert werden und anschließend in einem Simulationsprogramm für die Gesamtsystemsimulation integriert werden. Zur Durchführung dieser Variante ist es erforderlich, dass das Modell in einer für den Simulator verständlichen Form vorliegt. Im einfachsten Fall ist dem Simulationsprogramm die Modellierungssprache, die für die Modellierung eingesetzt wurde, bekannt. Da es sich allerdings oftmals um proprietäre Modellierungssprachen handelt, die mit dem Simulationsprogramm nicht kompatibel sind, muss der Modellcode in ein für das Simulationsprogramm verständliches Format konvertiert werden. Viele Programme bieten aus diesem Grund eine C-Code-Schnittstelle, die es einerseits ermöglicht, C-Code aus der softwarespezifischen Modellierungssprache zu generieren und andererseits diesen C-Code in die Zielsoftware zu importieren. Dieses Verfahren wird als Modellimport bezeichnet; hierfür wird zusätzlich zu den softwarespezifischen Modellierungssprachen ein Austauschformat benötigt, welches von den beteiligten Programmen generiert bzw. importiert werden kann. Ein Vorteil dieser Methode besteht darin, dass die am besten für den Anwendungsfall geeigneten Softwareprogramme eingesetzt werden können. Dem gegenüber steht der höhere Aufwand für den Modellimport. Ein weiterer Nachteil besteht darin, dass der generierte Code im Simulationsprogramm nicht geändert werden kann; der integrierte Modellcode stellt somit eine Blackbox dar. Bei Veränderung der Modellstruktur wird eine Veränderung am Ursprungsmodell und damit eine erneute Generierung des Quellcodes in z. B. C-Code erforderlich [118], [152].

3. Variante: Simulationskopplung

Der Grundgedanke zur Kopplung verschiedener Simulationsprogramme basiert darauf, dass für nahezu alle technischen Anwendungen spezifische Simulatoren verfügbar sind [129]. In der Simulationskopplung werden Teilsysteme in getrennten Simulatoren berechnet. An definierten Systemschnittstellen dieser Teilsimulationen werden zwischen einem oder mehreren Simulatoren uni- oder bidirektionale Ergebnisdaten ausgetauscht. Die Gesamtsystemsimulation setzt sich somit aus den getrennten Teilsystemsimulationen sowie dem Austausch der jeweiligen Simulationsergebnisse zusammen. Im Folgenden werden zwei Möglichkeiten der Simulationskopplung beschrieben.

Lose Simulationskopplung:

Ein Teil des zu simulierenden Gesamtsystems wird in einem Simulator berechnet. Die Ergebnisse dieser Berechnung dienen als Parameter einer anschließenden Teil- oder Gesamtsystemsimulation in einem oder mehreren Simulatoren. Die an der losen Kopplung beteiligten Simulationen tauschen erst nach Beendigung der alleinstehenden Einzelsimulationen Daten aus (siehe Abbildung 2-9). Diese Methode stellt eine sehr leicht zu implementierende Kopplung von Simulatoren dar, es ist allerdings nicht möglich, Daten während der Laufzeit auszutauschen. In Kapitel 5.2.3 wird die Umsetzung einer losen Kopplung zwischen zwei Simulationsprogrammen beschrieben. An diesem Beispiel wird verdeutlicht, wie ein automatisierter Parameterraustausch mithilfe eines standardisierten Austauschformats die Simulationskopplung vereinfacht und zusätzliche Übertragungsfehler der für die lose Kopplung benötigten Daten vermieden werden.

**Abbildung 2-9 Lose Kopplung von Simulatoren**Feste Simulationskopplung

Die Modellierung der Teilsysteme wird wie bei der zweiten Variante in unterschiedlichen Modellierungswerkzeugen vorgenommen. Allerdings ist das zu simulierende System wie auch bei der losen Kopplung auf eigenständige Simulationswerkzeuge aufgeteilt. Im Gegensatz zur

2.3 Einsatz von Simulation im Entwicklungsprozess

losen Kopplung werden in der festen Kopplung Variablen während der Laufzeit ausgetauscht. Der Vorteil dieser Methode gegenüber der losen Kopplung besteht darin, dass die Einzelsimulationen parallel ablaufen und sich gegenseitig während der Gesamtsimulationslaufzeit beeinflussen (siehe Abbildung 2-10). Hierdurch ist es möglich, dynamische Vorgänge eines Systems auf verteilten Simulatoren gekoppelt zu simulieren und immer das am besten geeignete Simulationsprogramm für den jeweiligen Teilsystemausschnitt zu benutzen. Für den Austausch der Variablen werden allerdings Programmschnittstellen zum Austausch der Variablen benötigt. Ein Nachteil dieser Methode besteht darin, dass der Aufwand für die Implementierung einer solchen Simulation den der losen Kopplung übersteigt. Anforderungen an die Synchronisation der Gesamtsimulation erhöhen den Aufwand zusätzlich. Zudem gestaltet sich die Handhabung einer Simulation aus gekoppelten Simulatoren durch das Fehlen einer einheitlichen Benutzeroberfläche für den Anwender im Allgemeinen als schwierig.

Einige Simulationsprogramme ermöglichen die feste Simulationskopplung mit anderen Simulationsprogrammen. Zum Beispiel wird von Saber, einem Simulationsprogramm für die Analyse elektrischer Netzwerke, eine Kosimulation mit Matlab/Simulink zur Verfügung gestellt. Diese Schnittstelle ermöglicht z. B. die Simulation eines elektrischen Systems mit Saber, während die Regelung dieses Systems in Matlab/Simulink durchgeführt wird.

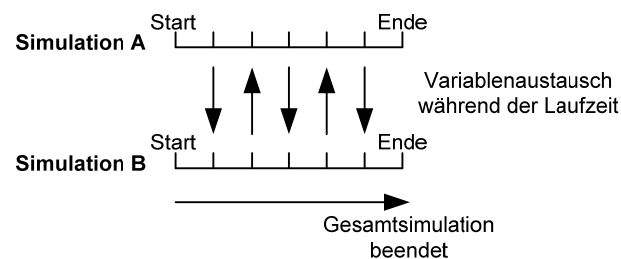


Abbildung 2-10 Feste Simulationskopplung

Ebenfalls in die Klasse der festen Simulatorkopplung fällt die verteilte Simulation. Das Ziel der verteilten Simulation ist es, unabhängige Ereignisse konkurrierend auszuführen und so die Simulationszeit zu beschleunigen [106]. Die Definition der verteilten Simulation lautet wie folgt [176]:

„Unter der verteilten Simulation soll die Simulation von Teilmodellen auf autonomen Simulatoren (d. h. mit Ereignissteuerung) verstanden werden, die Daten und Ergebnisse miteinander austauschen sowie gegenseitigen zeitlichen Restriktionen unterliegen (Zeitsynchronisation).“

Die verteilte Simulation ermöglicht zudem die räumliche Trennung der Simulationsteilnehmer auf Mehrrechnersystemen. Eine der am weitesten verbreiteten Architekturen für den Aufbau verteilter Anwendungen ist die CORBA-Technologie (engl.: Common Object Request Broker Architecture). CORBA eignet sich für plattformunabhängige Verteilung von Berechnungen [129]. In der TENT-Integrationsumgebung (Integrations- und Testumgebung für Simulation und Visualisierung) wurde CORBA zur Kopplung der auf unterschiedlichen Computersystemen laufenden Simulatoren eingesetzt. Neben der verteilten Simulation bietet TENT eine einheitliche Benutzeroberfläche, in der Simulationscodes, Pre- und Postprozessoren sowie Visualisierungstools kombiniert und konfiguriert werden können, und koordiniert die Kommunikation in verteilten Systemen, sodass rechenintensive Simulationen auf Hochleistungscomputern ablaufen können, während die Visualisierung der Ergebnisse auf Grafikworkstations erfolgt [123]. Mit TENT wurden bislang Fluid-Thermal- sowie Fluid-Struktur-Kopplungen im Flugzeugaufbau durchgeführt [42], [78].

Ein weiterer Standard (IEEE1516) für die verteilte Simulation ist die HLA (engl.: High Level Architecture). Ein Vorteil gegenüber CORBA liegt in der Fähigkeit eines umfassenden Zeitmanagements [159]. Dieses erlaubt die Synchronisation der Simulationszeitpunkte, zu denen Daten zwischen den Simulationsteilnehmern ausgetauscht werden [138]. Des Weiteren ermöglicht das Zeitmanagement, Simulationen in Echtzeit durchzuführen [53]. Zusätzlich ist die HLA im Hinblick auf Hard- und Software der eingesetzten Simulationsteilnehmer flexibler [80]. Durch den Einsatz der verteilten Simulation sowie auch durch die Modellseparation (vgl. Variante 4, Abbildung 2-8), ist es möglich, die Simulationszeit erheblich zu reduzieren. In Kapitel 5.2.4 wird eine verteilte Simulation auf Basis der HLA beschrieben.

In Abhängigkeit von Faktoren wie Werkzeugverfügbarkeit, Standort der Simulationsanwender, betrachteten Fachdisziplinen usw. ist die am besten geeignete Simulationsvariante für die Durchführung einer domänenübergreifenden Simulation auszuwählen. Es wurde gezeigt, dass es technisch möglich ist, Teilsysteme im Verbund zu simulieren, um deren Verhalten und Wechselwirkungen im Gesamtsystem virtuell zu untersuchen. Nicht betrachtet wurde dagegen die Frage, in welcher Form multidisziplinäre Simulationen sinnvoll und effektiv in den Entwicklungsprozess eingebunden werden können. Ebenso müssen im weiteren Verlauf dieser Arbeit die Voraussetzungen für die Integration multidisziplinärer Simulationen im systemübergreifenden Entwicklungsprozess untersucht werden. Aus diesem Grund werden im Kapitel 4 verschiedene Konzeptalternativen entworfen und anschließend auf Erfüllung der Anforderungen aus Kapitel 3 hin überprüft.

2.3.3 Einsatz von Simulation in der Automobilindustrie

Seit der Einführung der computergestützten Berechnungen im Fahrzeugbau nimmt die Automobilindustrie eine Vorreiterrolle bei der Entwicklung und Nutzung von rechnergestützten Werkzeugen ein. Beginnend mit der Einführung der computergestützten Konstruktion (Abk. CAD; engl.: Computer Aided Design) über den Einsatz von Simulationstechniken bis hin zur digitalen Fabrik. Die hohe Akzeptanz im Hinblick auf den Einsatz von Simulationsprogrammen in der Automobilindustrie darf allerdings nicht über bestehende Schwachstellen hinwegtäuschen. So erschweren fehlende Standards den effizienten Einsatz der Simulationstechnik. Zusätzlich ist das Wissen über Simulationsmethoden zu wenig verbreitet, und der Einsatz ist stark segmentiert und von einer Durchgängigkeit in der Produktentwicklung weit entfernt [9], [154].

Zusammen mit der Luft- und Raumfahrtindustrie prägt der Autobau die kommerzielle Simulationswerkzeugentwicklung. Dies ist darin begründet, dass es sich hierbei um eine der wenigen Branchen handelt, die aufgrund ihrer umfassenden Aufgabenstellung in der Lage ist, eine durchgängige Integration solcher Werkzeuge für den gesamten Produktlebenszyklus umzusetzen. Die Umsetzung einer durchgängigen Simulationsunterstützung ist nicht durch die Anpassung der Unternehmensprozesse an die Informationstechnologie geprägt; viel wichtiger ist, dass die Mitarbeiter durch den Einsatz der Technologie einen eigenen Nutzen erwarten. Aus diesem Grund ist der durchgängige Einsatz von Simulationstechnologien sowie die damit verbundene konsistente Bereitstellung von Daten und Informationen eine strategische Entscheidung, die gelebt werden muss [9].

In diesem Kapitel wird der Einsatz von Simulationstechniken in der Automobilindustrie beschrieben. Dafür werden zunächst verschiedene Simulationsanwendungen beschrieben, um einen Überblick über den derzeitigen Einsatz von Simulationstechniken in diesem Industriezweig zu erhalten. Des Weiteren wird auf vorhandene Informationsbereitstellungstechnologien eingegangen. Abschließend wird begründet, warum Methodiken der Automobilindustrie nicht eins zu eins in der Luftfahrtindustrie übernommen werden können.

In der Automobilindustrie kommen sämtliche im Kapitel 2.1 beschriebenen Simulationstechniken in nahezu allen Bereichen des Fahrzeugbaus zum Einsatz [180], [181]. So werden für die Entwicklung der Fahrzeugtür, eines der komplexesten Bauteile der Fahrzeugkarosserie, Finite-Elemente-Simulationen durchgeführt, um z. B. die statische Fensterrahmensteifigkeit oder das dynamische Crashverhalten zu ermitteln. Ohne die Zuhilfenahme dieser rechnergestützten Berechnungsmethoden wäre die effiziente Entwicklung und Konstruktion eines solchen Bauteils schon heute nicht mehr möglich [145]. Des Weiteren werden FEM für die Optimierung der Karosseriestruktur eingesetzt. Für die akustische Optimierung von Seitenspiegeln werden CFD-Simulationen eingesetzt, um so die geräuschärmste Geometrie zu ermitteln [58]. Für die Analyse des thermischen Verhaltens von Brems Scheiben wird ein gekoppelter Simulationsansatz, bestehend aus CFD- und FEM-Simulation, verwendet [83]. Im Automobilbau kommen Strömungssimulationen in vielen weiteren Bereichen, wie z. B. der aerodynamischen Optimierung der Karosserie, der Strahlungsanalyse von Scheinwerfern usw., zum Einsatz.

Um das Verhalten von Hardwarekomponenten noch vor der Fertigstellung des Gesamtproduktes analysieren zu können, werden Hardware in the Loop-Simulationen eingesetzt. So ist es z. B. möglich, Steuergeräte als echte Hardwarekomponenten zu betreiben, während das mechanische Fahrzeug und die Leistungselektronik des Antriebs in Echtzeit auf einem Computer simuliert werden. Damit ist es möglich, die Fahrdynamik eines Pkws noch vor der Fertigstellung des Gesamtproduktes zu analysieren [142], [154].

2.3 Einsatz von Simulation im Entwicklungsprozess

Neben der gestaltorientierten und den HIL-Simulationen wird die Systemsimulation ebenfalls in der Automobilindustrie eingesetzt. In den meisten Fällen wird sie für die Reglerentwicklung und -analyse sowie zur Analyse der Fahrdynamik eingesetzt. Das Simulationsprogramm Matlab/Simulink wird z. B. für die Berechnung des thermischen Netzwerkes von Motoren eingesetzt [133]. Ein weiteres Einsatzgebiet der Systemsimulatoren ist die Berechnung von Fluidströmen sowie die Berechnung chemischer Prozesse mithilfe von verfahrenstechnischen Simulatoren wie Aspen Plus des Unternehmens AspenTech. In diesem Kapitel wird ein Überblick des derzeitigen Einsatzes der Simulationstechnik in der Automobilindustrie gegeben (vgl. Tabelle 2-3), für weitere Informationen zu diesem Thema sei an dieser Stelle auf folgende Quellen verwiesen [9], [181], [180].

Anwendungsgebiet	Simulationstyp
<ul style="list-style-type: none"> - Fahrzeugtür - Längenoptimierung von Bremsschläuchen - Schweißvorgänge - Crashverhalten - Strukturoptimierung - Werkstoffanalyse usw. 	FEM gestaltorientierte Simulation
<ul style="list-style-type: none"> - Akustische Auslegung von Bauteilen - Aerodynamische Optimierung - Scheinwerferanalyse usw. 	CFD-Simulation
<ul style="list-style-type: none"> - Fahrdynamik - Reglerentwurf - Logiksimulation von Bussystemen - Analyse des elektrischen Bordnetzes 	Systemsimulation (signalfussorientiert) (zeitdiskret)

Tabelle 2-3 Simulationstechnik in der Automobilindustrie

Im Autobau wurde sich bereits mit dem Thema der durchgängigen Simulationsunterstützung in der Produktentwicklung beschäftigt. Der Autobauer BMW und die ITK GmbH entwickelten die durchgängige Simulationsumgebung ISAR (Integrierte Simulationsumgebung für Fahrdynamik mit Regelsystemen) zur Entwicklung und Absicherung von fahrdynamischen Regelsystemen. Im Folgenden wird diese Simulationsumgebung näher beschrieben.

Signifikante Fortschritte in der Fahrdynamik werden beim heutigen Stand der Technik nur durch die Integration von elektronischen Systemen erreicht. Aktuell werden Systeme wie z. B. ABS, ESP, Dynamic Drive, Bremsassistentz und andere Assistenzsysteme eingesetzt. Diese Systeme erlauben die Optimierung im Bezug auf Fahrdynamik, Sicherheit und Komfort. Der verstärkte Einsatz dieser Technologien erhöht allerdings die Komplexität des Gesamtsystems Auto. Aus diesem Grund wird für die Entwicklung hochintegrierter Regelsysteme ein integrierter Prozess, der mit der Vorentwicklung beginnt und mit der Absicherung der Applikation endet, benötigt. Da die Steuergeräte im Hinblick auf die Gesamtfahrzeugeigenschaften entwickelt werden, muss die strukturelle und funktionale Komplexität schrittweise und durchgängig aufgebaut sein (vgl. Abbildung 2-11).

In der Konzeptphase werden die Hauptkomponenten des Regelsystems berücksichtigt. In der Simulation werden dabei Model-in-the-Loop (Abk. MIL)-Analysen durchgeführt. Dabei werden Prinzipmodelle der Regelsysteme in einem Fahrzeugkonzeptmodell untersucht, um so erste Aussagen in den Bereichen Fahrdynamik, Sicherheit und Komfort treffen zu können. In der anschließenden Spezifikationsphase werden detaillierte Funktionsmodelle der Subsysteme entwickelt. Ziel ist es, die Funktionalität der Regelalgorithmen zu untersuchen. Dafür wird in einer Software-in-the-Loop-Simulation (Abk. SIL) das Verhalten des Fahrzeuggesamtsystems simuliert. In der folgenden Absicherungsphase vollzieht sich die schrittweise Entwicklung der Hardwarekomponenten. Um die Funktionalität der Komponenten zu überprüfen, werden diese in skalierbaren Hardware in the Loop-Prüfständen getestet. Hierfür sind echtzeitfähige Modelle notwendig. In der abschließenden Phase erfolgt die Überprüfung der Steuergeräte vorwiegend während realer Testfahrten.

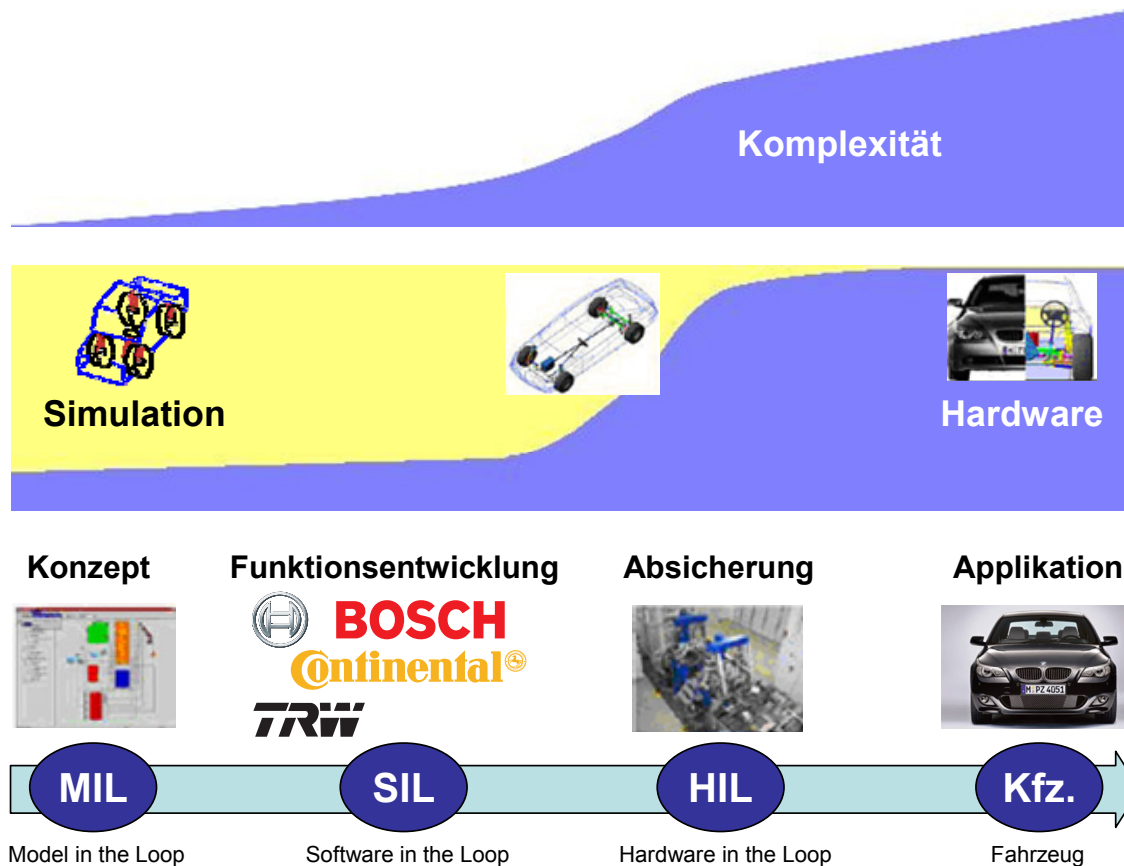


Abbildung 2-11 Schrittweiser Übergang von der virtuellen zur Hardwareentwicklung [91]

Ein erfolgreicher Prozess für eine hochintegrierte Systementwicklung von mechanischen Fahrzeug und mechatronischen Systemen beinhaltet

- einen schrittweisen Aufbau der Komplexität,
- einen schrittweisen Übergang von virtueller Entwicklung zu realen Tests.

Für die Unterstützung dieses Prozesses muss das Simulationswerkzeug modular und übergreifend über die Funktion und den Prozess sein sowie eine schnelle und flexible Anwendung gewährleisten [91]. An diesem Beispiel wird deutlich, dass bereits Konzepte für die durchgängige Simulationsunterstützung in der Automobilindustrie existieren. Allerdings ist die ISAR-Simulationsumgebung speziell für die Entwicklung von Reglersystemen entworfen worden und somit nicht ohne Weiteres auf andere Bereiche anwendbar. Das prinzipielle Vorgehen sowie die beschriebenen Anforderungen an ein durchgängiges Simulationskonzept besitzen ebenfalls Gültigkeit für ein zukünftiges Konzept zur Simulationsunterstützung in der Entwicklung von Flugzeugsystemen und müssen daher in der Konzeptentwicklung (vgl. Kapitel 4) berücksichtigt werden.

Neben den bereits angesprochenen Einsatzgebieten der Simulationstechnik im Automobilbau wird die Simulation in der Planung und Auslegung neuer Produktionsanlagen, für die Optimierung von Fertigungsstraßen, in der Betriebsführung sowie zur Verbesserung von logistischen Prozessen in der Automobilindustrie eingesetzt [9]. In den letzten Jahren wurde der Begriff des Simultaneous Engineering geprägt (Abk. SE; deutsch: gleichzeitige Entwicklung). Grundgedanke dieses Verfahrens ist die zeitliche Überlappung von eigentlich aufeinanderfolgenden Arbeitsabläufen. Besonders nutzbringend ist der Einsatz dieser Technik zwischen den traditionell streng getrennten, sequenziellen Arbeitsabläufen der Produktentwicklung und Produktionsmittelplanung. Durch den Einsatz der Simulationstechnologie ist es möglich, bereits während der Produktentwicklung die benötigten Fertigungsanlagen zu entwickeln. In der Automobilbranche geht das Simultaneous Engineering soweit, dass bereits vor der Fertigstellung des Produktes ganze Produktionswerke aufgebaut werden.

Simulationstechniken kommen, wie im Vorangegangenen beschrieben, in nahezu allen Bereichen der Automobilindustrie zum Einsatz. Von einer Durchgängigkeit der

2.3 Einsatz von Simulation im Entwicklungsprozess

Simulations- und Berechnungsdaten ist man allerdings weit entfernt. Diese Tatsache ist nicht zuletzt auf die unzureichende Berücksichtigung der Simulationsdaten in den Produktdatenmanagementsystemen (PDM-Systemen) zurückzuführen. Zwar gibt es Ansätze, die Daten einer Entwicklung komplett in einem computerbasierten Datenmodell abzubilden. Durchgeführt wurde dies im Fahrzeugbau bei der Entwicklung des Ford Mondeo mit dem C3P-System. Allerdings beschränkte sich dieses Projekt auf die Bereitstellung von Geometriedaten, nicht auf die Bereitstellung sämtlicher Entwicklungsdaten [157]. Merkt schreibt, dass 51 % der für eine Simulationsstudie benötigten Zeit allein für die Datenbeschaffung veranschlagt werden [108]. Damit ist die Datenbeschaffung im Vergleich zur eigentlichen wertschöpfenden Tätigkeit sehr hoch. Aus diesem Grund wurde 2003 die Projektgruppe SimPDM – Integration der Simulation und Berechnung in eine PDM-Umgebung –, bestehend aus führenden deutschen Autobauern, gegründet. Ihre Zielsetzung ist es, zukünftig die Computer Aided Engineering-Systeme (CAE) mittels der PDM-Technologien synchron an den Entwicklungsprozess zu koppeln und den Datenaustausch zu unterstützen.

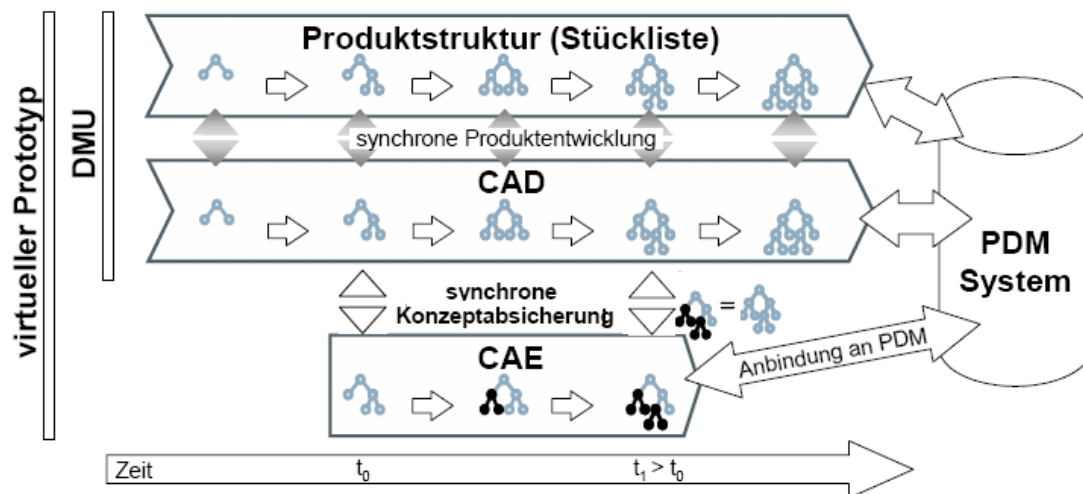


Abbildung 2-12 Ziel einer synchronen Kopplung von CAE- und PDM-Systemen [108]

In Abbildung 2-12 ist die Kopplung von CAE- und PDM-Systemen dargestellt. Wie im Vorangegangenen erwähnt, werden die Geometriedaten (CAD-Daten) bereits heute in PDM-Systemen hinterlegt und ermöglichen so zusammen mit der Produktstruktur den Aufbau eines digitalen Versuchsmodells (Abk. DMU; engl.: Digital Mock-Up). Für die Erstellung eines virtuellen Prototyps werden zusätzlich die Daten der Simulation und andere CAE-Systeme benötigt.

Durch den Einsatz von Simulationsdatenmanagementsystemen wie, z. B. dem Sim-Manager von TI Automotive, konnten bereits Kostenreduzierungen von 25 % im Bereich der Prototypenentwicklung und der Tests erreicht werden [17].

„Die aktuelle Entwicklung im Kontext der digitalen Produktion und der digitalen Fabrik verdeutlicht die Intention, alle am Planungsprozess beteiligten Akteure zusammenzuführen und über die Verzahnung der einzelnen Planungsschritte die Umsetzung eines Simultaneous Virtual Engineering zu erreichen [9].“

Die Automobil- und Flugzeugindustrie weisen viele Gemeinsamkeiten, aber auch gravierende Unterschiede auf, sodass es nicht immer möglich ist, Technologien, Methodiken und Prozesse zu transferieren. Im Einzelfall müssen diese adaptiert und um zusätzliche Anforderungen erweitert werden. In beiden Industriezweigen herrscht ein starker Wettbewerb, der durch eine hohe Marktsättigung noch verstärkt wird. Um konkurrenzfähig zu bleiben, müssen die Entwicklungskosten bei gleichzeitig steigender Qualität reduziert werden. Allerdings ist die Automobilindustrie viel stärker kundenorientiert. Zudem zeichnet sich das Beschaffungswesen (engl.: procurement) in der Automobilindustrie dadurch aus, dass aus vielen Zulieferern, die bereits hohe Ausgaben für Produktentwicklungskosten geleistet haben, nur ein Zulieferer ausgewählt wird. Die Automobilindustrie beteiligt sich nicht an den bereits getätigten Entwicklungskosten, dieses ist einzig und allein die Aufgabe der Zulieferer. Somit tritt der Zulieferer in Vorkasse, ohne zu wissen, ob er für die Zulieferung eines Produktes ausge-

wählt wird. In der Luftfahrt werden Forschungsprojekte zusammen mit Projektpartnern, die in den meisten Fällen später als Zulieferer eingesetzt werden, durchgeführt [51].

Ein weiteres Unterscheidungsmerkmal sind die wesentlich höheren Sicherheitsanforderungen in der Luftfahrt. Die für die Aufrechterhaltung des Flugbetriebs verantwortlichen Systeme an Bord eines Flugzeugs müssen redundant ausgelegt sein (z. B. der Bordrechner). Diesen strengen Anforderungen unterliegen auch Softwareprogramme, die an Bord des Flugzeugs oder aber auch in der Entwicklung eingesetzt werden. Wird z. B. der Code eines Reglers von einem Simulationsprogramm generiert, muss sichergestellt werden, dass dieser Code ebenfalls gewisse Anforderungen erfüllt. Das Programm SCADE von Esterel erlaubt z. B. die zertifizierte Codegenerierung in der Luftfahrtindustrie [47]. In der folgenden Tabelle sind weitere Unterschiede zwischen der Automobil- und der Luftfahrtindustrie dargestellt.

	Auto [VW New Beetle]	Flugzeug	Einheit
Entwicklungszyklen	3,5	4,5	Jahre
Komplexität	10.000	130.000	Einzelteile
Beteiligte	800	6.800	Personen
Stückzahl	100.000	50	Einheiten/Jahr
Produktlaufzeit	6	30	Jahre
Preis	17.000	130000000	\$
Zulieferer	800	10.000	Anzahl
Entwicklungskosten	500.000.000	3.000.000.000	\$

Tabelle 2-4 Produktionsaufwand verschiedener Eigenschaften in der Automobil- und Luftfahrtindustrie [172] [148]

Zusammenfassend lässt sich sagen, dass auch im Automobilbau kein Standard für den Einsatz von Simulationstechniken existiert [9], [91]. Des Weiteren ist es aufgrund der beschriebenen Unterschiede zwischen der Luftfahrt- und der Automobilindustrie nicht immer möglich, Prozesse, Methodiken und Verfahrensweisen der Automobilindustrie eins zu eins in die Luftfahrt zu übernehmen. Für die Entwicklung einer durchgängigen Simulationsunterstützung in der Luftfahrt können jedoch die in diesem Kapitel vorgestellten Methoden als Basis für ein verbessertes Vorgehensmodell dienen. Für einen umfassenden Überblick des derzeitigen Stands der virtuellen Produktentstehung in der Automobilindustrie sei an dieser Stelle auf folgende Quellen verwiesen [154].

2.3.4 Einsatz von Simulation in der Luftfahrt

Im Folgenden wird am Beispiel des Flugzeugherstellers Airbus der Simulationseinsatz in der Flugzeugsystementwicklung erläutert.

In den Unternehmensdokumenten wird der Einsatz von Simulation zwar beschrieben, aber nicht verbindlich vorgeschrieben. Die Anwendung von Simulation in der Entwicklung ist den Besonderheiten der jeweiligen Entwicklungsaufgabe anzupassen. In Abbildung 2-13 ist der Simulationseinsatz in den verschiedenen Produktentwicklungsphasen dargestellt.

Der Prozess ist in fünf Phasen und 14 Meilensteine eingeteilt, wobei Simulationstechniken unterstützend in der Konzept-, Definitions- und Entwicklungsphase eingesetzt werden. In der Konzeptphase wird die Simulation zur Analyse des Systems vorgeschlagen. In der anschließenden Definitionsphase soll eine funktionale System-simulation durchgeführt werden, die im Zuge der Entwicklung weiter detailliert wird. Simulation ist gemäß der Airbus-Richtlinien entlang des Entwicklungsprozesses zur Validierung und Verifizierung (Abk. V&V) einzusetzen. Schließlich ist die Integration der Systeme (M9) mit Simulation zu unterstützen.

Zusammenfassend lässt sich sagen, dass Richtlinien bezüglich der Frage, wann Simulation im Prozess eingesetzt werden sollte, existieren, es handelt sich dabei allerdings nicht um konkrete Handlungsvorgaben. Fragen zur konkreten Umsetzung der Richtlinien, wie zum Beispiel zur Form der Datenbereitstellung im Prozess, werden nicht beantwortet.

In einer multidisziplinären Produktentwicklung müssen Wechselwirkungen und Inkompatibilitäten der einzelnen Teilsysteme frühzeitig analysiert werden. Aus diesem Grund ist es sinnvoll, ganze Systeme oder Teilsysteme in einer gemeinsamen virtuellen Umgebung zu simulieren, um das Gesamtverhalten des betrachteten Bereichs zu überprüfen.

2.3 Einsatz von Simulation im Entwicklungsprozess

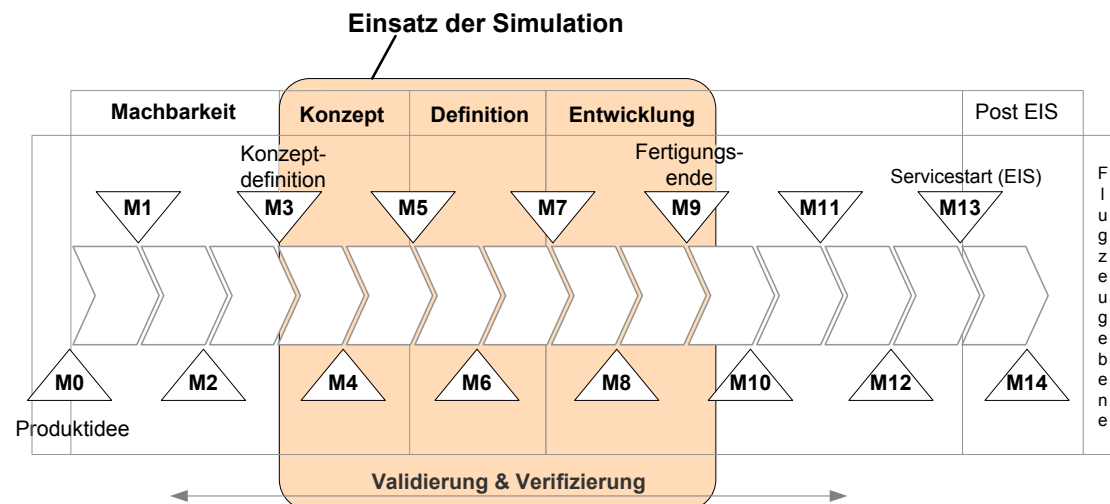


Abbildung 2-13 Simulation in der Entwicklung bei Airbus [5]

Bedingt durch dieses Vorgehen, können Fehler bereits in den frühen Entwicklungsphasen erkannt und damit kostenintensive Änderungsschleifen vermieden werden. Erste Integrationstests würden so früher, nämlich in einer virtuellen Umgebung mit virtuellen Komponenten, stattfinden, sodass Änderungen bereits in den frühen Phasen der Entwicklung vorgenommen werden könnten. Realisiert werden kann dies, indem einzelne Module erstellt und in eine gemeinsame Simulationsumgebung eingebunden werden. Über Schnittstellen findet der Austausch mit Modellen anderer Module statt. Die virtuellen Modelle werden im Lauf der Entwicklung durch reale Software und Hardware ersetzt. Voraussetzungen für eine virtuelle Gesamtbetrachtung sind eine geeignete Simulationsumgebung, das Einhalten von Schnittstellenvorgaben, die Anpassungsfähigkeit an verschiedene Detaillierungsgrade, die Wiederverwendbarkeit usw. [188], [77], [81], [139].

Die Bausteine zur Realisierung einer systemübergreifenden Simulation sind zum Teil vorhanden. Die technische Umsetzung der Simulationsumgebung kann z. B. mithilfe von Simulationskopplungen erreicht werden (vgl. Kapitel 4.1). Für die Bereitstellung von Schnittstellenstandards und Dokumentationsrichtlinien existieren bereits Lösungen, die im Kapitel 2.4 vorgestellt werden. Es fehlt allerdings ein gesamtes Prozessmodell, das den Zeitpunkt sowie den konkreten Einsatz der Simulationstechnik im Entwicklungsprozess festlegt.

2.4 Informationsbereitstellung

Ein wichtiger Aspekt für die erfolgreiche Produktentwicklung ist die Bereitstellung von Informationen. Besonders in der Entwicklung von hochintegrierten Flugzeugsystemen stellt sich die Frage, wie die erforderlichen Informationen bzw. Daten und Modelle und damit das Wissen im Prozess bereitgestellt werden können.

Wissen ist bekanntlich der einzige Rohstoff, der sich beim Teilen vermehrt [74]. Aus diesem Grund ist es von großer Wichtigkeit, das in der Entwicklung erlangte Wissen allen Beteiligten zugänglich zu machen. Hierbei muss darauf geachtet werden, dass jeweils nur die benötigten Informationen zur Verfügung gestellt werden. In diesem Abschnitt werden Anforderungen an die Informationsbereitstellung beschreiben, anschließend werden Möglichkeiten zur Bereitstellung von Informationen in der Entwicklung dargestellt.

2.4.1 Anforderungen an die Informationsbereitstellung in der Produktentwicklung

Bei der Entwicklung komplexer Produkte besteht die Schwierigkeit darin, dass die an der Entwicklung Beteiligten erst verspätet oder nach mehreren Iterationszyklen den gleichen Wissensstand haben, da sie häufig getrennt voneinander den Entwicklungsprozess durchlaufen. Die im Prozess erforderlichen Daten müssen mit großem Aufwand gesammelt und kommuniziert werden, sodass es häufig zu Missverständnissen, Interpretationsunterschieden und Ungenauigkeiten kommt. Die Entwicklungszeit und damit auch die -kosten hängen zunehmend vom Vorhandensein digitaler Informationen ab [147], [157], [87], [192], [193].

Die Zunahme der Rechnerunterstützung in der Produktentwicklung, bedingt durch den Einsatz von CAX-Systemen (engl.: Computer Aided x, wobei das x ein Platzhalter für ver-

schiedene Buchstaben darstellt, z. B. Engineering, Design, Manufacturing, usw.), führt zu einem Anstieg der digitalen Entwicklungsdaten. Hieraus resultieren Konsistenz-, Redundanz-, Transparenz- und Aktualitätsprobleme. Nach Vanja entstehen die meisten Schwierigkeiten in der Produktentwicklung durch eine unzureichende Verwaltung von Daten und Dokumenten, bei der es schwierig ist, die jeweils aktuellen und gültigen Informationen zu lokalisieren [173].

In der Flugzeugproduktion nehmen die individuellen Kundenwünsche einen hohen Stellenwert ein. Aus diesem Grund wird ein stabiles und widerspruchsfreies Konfigurationsmanagement auch als „einer der kritischen Erfolgsfaktoren für einen effizienten Produktentstehungsprozess von Verkehrsflugzeugen“ bezeichnet [147]. Folgende Anforderungen müssen von einem Informationsmanagementsystem erfüllt werden [173], [87], [193], [143], [179].

- Bereitstellung aller erforderlichen Informationen
- Ermöglichen eines frühzeitigen Ergebnisaustauschs und -abgleichs
- Aktueller und konsistenter Daten- bzw. Wissensbestand
- Standardisierte Schnittstellen (in digitalen Werkzeugen sowie zum Modellaustausch)
- Durchgängige Unterstützung dynamischer, unternehmensübergreifender Prozessflüsse
- Vermeidung redundanter Daten
- Bereitstellung eines Konfigurationsmanagements

Basierend auf den Anforderungen an ein Informationsmanagementsystem werden im Folgenden Lösungsmöglichkeiten beschrieben, die diese Anforderungen erfüllen.

2.4.2 Möglichkeiten der Informationsbereitstellung

Durch den rechnerunterstützten Entwicklungsprozess müssen in den Unternehmen bereits große Datensätze verwaltet werden. Um der Anforderung zur Bereitstellung aller erforderlichen Information nachkommen zu können, bestehen die Möglichkeiten des Zusammenführens bestehender Verwaltungsprogramme oder der Substitution durch ein neues Programm. Die Schwierigkeiten beim Wechsel auf ein einziges Programm liegen im kostenintensiven Wechsel auf die neue Programmstruktur sowie in der Tatsache, dass die Gesamtlösung den Standard der vorherigen Teillösungen erreicht [45].

Nach Schäppi besteht keine Notwendigkeit, alle Bereiche eines Unternehmens mit einem einzigen System zu unterstützen, da ca. 80 % der Daten in den jeweiligen Bereichen erzeugt, bearbeitet, verwendet und dort gespeichert werden [147]. In diesem Zusammenhang werden allerdings nicht die besonderen Bedürfnisse eines abteilungsübergreifenden Datenaustausches im systemübergreifenden Entwicklungsprozess berücksichtigt. Dennoch wird deutlich, dass nicht eine Substitution bestehender Systeme zur Datenspeicherung genügt, sondern eine Einbindung des Informationsmanagements in den Prozess nötig wird [188].

In einer verteilten bzw. dezentralen Entwicklung bietet es sich an, den Datenaustausch über eine einheitliche Plattform zu organisieren und integrierte Systemdienste, wie Konfigurationsmanagement, Rechteverwaltung, Datensicherheitsdienste und Produktdatenmanagement zu steuern. Dies kann mithilfe des Client-Server-Prinzips realisiert werden. Daten können zentral auf einem Server abgelegt werden und stehen allen Beteiligten während des gesamten Prozesses zur Verfügung. Als Resultat der gemeinsamen Datennutzung ist eine Reduzierung der Entwicklungszeit und -kosten zu nennen [87], [124], [143], [147]. Auch bei einer dezentralen Datenbereitstellung können die Informationen zentral in einem virtuellen Projektraum bereitgestellt werden [188].

Nach Vanja gehört die EDM/PDM-Technologie (engl.: Engineering Data Management/Product Data Management) zu den wichtigsten Voraussetzungen für die erfolgreiche Produktentwicklung [173]. Das EDM-System betrifft den Arbeitsprozess und die ihn beschreibenden Daten, während das PDM-System die produktbeschreibenden Strukturen, Dokumente und Daten umfasst. Diese beiden Schichten sind notwendig, um eine integrative Rechnerunterstützung in der Produktentwicklung zu ermöglichen. Ein EDM/PDM-System unterstützt alle Aktivitäten bezüglich Erstellung, Änderung und Archivierung einer Produktstruktur. Zusätzlich verwaltet es alle Entwicklungsdokumente und -daten, die von den CAX-Systemen bereitgestellt werden. In Abbildung 2-14 ist das Zusammenspiel eines EDM/PDM-Systems mit anderen Systemen der Datenverarbeitung dargestellt.

2.4 Informationsbereitstellung

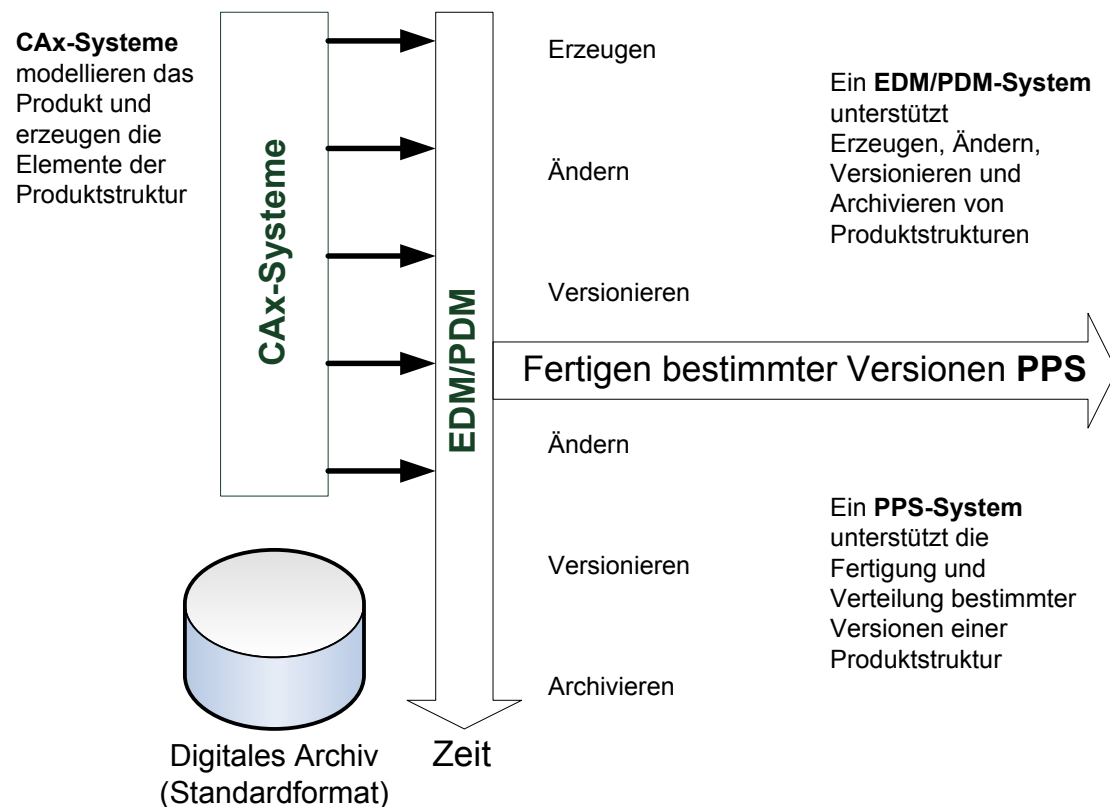


Abbildung 2-14 Zusammenspiel von CAx-, EDM/PDM- und PPS-Systemen [182]

Mithilfe des PPS-Systems (Produktplanung und Steuerung) werden Fertigungen bestimmter Versionen der Produktstruktur umgesetzt. Es wird deutlich, dass das EDM/PDM-System das informationstechnische Rückgrat der Produktentwicklung darstellt. Durch den Einsatz eines EDM/PDM-Systems werden folgende Vorteile erreicht [173]:

- Entwicklungszeit bis zur Marktreife wird verkürzt
- Erhöhung der Integration, Transparenz und Konsistenz der in der Produktentwicklung bearbeiteten Daten
- Vereinfachung des Informationszugriffs
- Reduzierung der Entwicklungskosten
- Digitale Archivierung (Reduzierung von Papierunterlagen)
- Konfigurationsmanagement
- Vollständige Bereitstellung aller Archivdaten

In Tabelle 2-5 sind die gemittelten Nutzen von Unternehmen, die bereits EDM/PDM-Systeme verwenden, dargestellt.

Kriterium	Reduktion um mindestens
Kostensituation im Engineering	15 %
Anzahl der Entwicklungszyklen	25 %
Zeitaufwand für das Änderungswesen	30 %
Anzahl der Änderungen	40 %

Tabelle 2-5 Aufwandsreduzierung durch den Einsatz von EDM/PDM-Systemen [173]

Die Einführung eines EDM/PDM-Systems sollte stets unternehmensweit erfolgen. Es ist dabei zu beachten, dass die Einführung kosten- und zeitintensiv ist. Als Leitfaden für die Einführung eines EDM/PDM-Systems in einem Unternehmen dient die VDI-Richtlinie 2219 [182]. Für den Aufbau und die konkrete Umsetzung eines EDM/PDM-Systems sei an dieser Stelle auf folgende Quellen verwiesen [147], [173], [182].

Zusammenfassend kann gesagt werden, dass es von Vorteil ist, spezialisierte Programme beizubehalten und diese nicht durch ein einziges Programm zu ersetzen. Es wurde gezeigt, dass eine zentrale Bereitstellung der Informationen (z. B. mithilfe des Client-Server-

Prinzips) erreicht werden kann. Dieses ist besonders für das Informationsmanagement in domänenübergreifenden Entwicklungen von Bedeutung.

Wie bereits im vorangegangenen Abschnitt beschrieben wurde, trägt die Wiederverwendung von Modellen und Simulationsergebnissen zur Reduzierung der Entwicklungszeit bei. Im Anhang (vgl. Kapitel 7.4) sind aus diesem Grund verschiedene Programme zum Austausch und zur Verwaltung von Simulationsdaten in der Entwicklung beschrieben. Zudem werden standardisierte Datenformate, die sich für den Austausch von Simulationsdaten eignen, dargestellt. In Kapitel 5.3.1 wird eine lose Simulationskopplung für den automatischen Datenaustausch zwischen Simulationsprogrammen, basierend auf einem standardisierten XML-Format, beschrieben.

In Kapitel 2 wurden wichtige Grundlagen, die für das weitere Verständnis der Arbeit von Interesse sind, bereitgestellt. Zu Beginn wurden verschiedenen Simulationstechniken und Programme beschrieben. Im Anschluss daran wurden Methodiken und Verfahren der Produktentwicklung vorgestellt, dabei wurde auf den Einsatz in Bezug auf eine multidisziplinäre Produktentwicklung eingegangen. Es wurde beschrieben, wie Simulationstechniken zum gegenwärtigen Zeitpunkt in verschiedenen Industriezweigen eingesetzt werden. Im folgenden Kapitel wird der Entwicklungsprozess in der Luftfahrtindustrie analysiert. Aus den ermittelten Defiziten der Istsituation werden anschließend Anforderungen an ein neues simulationsgestütztes Entwicklungskonzept generiert.

3 Analyse des Einsatzes von Simulation im Entwicklungsprozess von Flugzeugsystemen

Das Ziel dieser Arbeit ist die Entwicklung eines Konzeptes zur durchgängigen Simulationsunterstützung in der Entwicklung von komplexen und hochintegrierten Produkten. Der Simulationseinsatz soll dabei zu einer effektiveren und effizienteren Gestaltung des bestehenden Entwicklungsprozesses führen. Aus diesem Grund wurden im vorangegangenen Kapitel die theoretischen Grundlagen für allgemeingültige Vorgehensmodelle dargestellt. Es wurde darauf hingewiesen, dass zurzeit eine konkrete Vorgehensmethodik für die integrierte Umsetzung zum Einsatz von Simulationstechniken und zur Informationsbereitstellung im Entwicklungsprozess fehlt [86].

Basierend auf den in Kapitel 2 vorgestellten Grundlagen wird in diesem Kapitel die Istsituation im Flugzeugsystementwicklungsprozess analysiert, um die derzeitige Defizite zu identifizieren. Zusätzlich zur literaturbasierten Analyse wurden am Entwicklungsprozess beteiligte Experten über das derzeitige Vorgehen, den Einsatz von Simulation sowie Prozessdefizite in der Entwicklung von Flugzeugsystemen befragt. Die ermittelten Defizite dienen im Rahmen der Konzeptentwicklung zur Herleitung von Anforderungen an ein zukünftiges simulationsgestütztes Konzept.

3.1 Istsituation der Systementwicklung

In diesem Kapitel wird der derzeitige Produktentstehungsprozess in der Luftfahrtindustrie dargestellt. Die Erkenntnisse basieren auf den in den Kapiteln 2.2 und 2.3 vorgestellten Grundlagen, herstellertypischen Richtlinien sowie auf Expertengesprächen, die in Zusammenarbeit mit Airbus durchgeführt wurden. Es wurden an der Entwicklung beteiligte Experten verschiedener Systemabteilungen und EADS-Unternehmenseinheiten an verschiedenen europäischen Standorten befragt, um so einen ganzheitlichen Eindruck des momentanen Systementwicklungsprozesses zu erhalten.

Die Entwicklung eines Flugzeugs zeichnet sich durch eine hohe Produktkomplexität sowie hohe Anforderungen an die Sicherheit aus. Es müssen Interaktionen verschiedener physikalischer Disziplinen in einer multinationalen, verteilten Unternehmensstruktur berücksichtigt werden. Zusätzlich zu den Herausforderungen, die eine Flugzeugentwicklung an alle Beteiligten stellt, existieren kulturelle und kommunikative Unterschiede. Ein klar strukturierter Prozess ist für eine erfolgreiche Flugzeugentwicklung von besonderer Bedeutung. Wie bereits in Kapitel 2.2.2 beschrieben wurde, müssen hierfür verschiedene Ziele, sogenannte Meilensteine, sukzessive abgearbeitet werden. In Abbildung 3-1 ist der Airbus Meilensteinplan dargestellt.

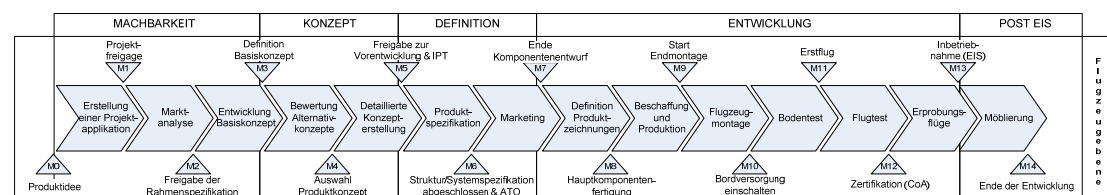


Abbildung 3-1 Meilensteinplan im Flugzeugbau [5]

Eine hierarchische Aufteilung des Gesamtsystems dient zur Einteilung der Flugzeugsysteme in die sogenannten ATA-Kapitel (engl.: Air Transport Association) [119]. Für die einzelnen Flugzeugsysteme (z. B. Hydraulik, Elektrik, Klimaanlage usw.) wird der Systementwicklungsprozess gemäß Abbildung 3-2 durchlaufen (vgl. Abbildung 2-4). Der Entwicklungsprozess wird in der weiteren Beschreibung in die vier Hauptphasen Konzept, Detaillierung, Komponentenentwicklung sowie Test unterteilt.

Im linken Strang des V-Modells werden die Anforderungs- und Spezifikationsdokumente generiert. Dabei wird nach dem Top-down-Prinzip verfahren, d. h. die übergeordneten Flugzeuganforderungen werden bis auf die Komponentenebene verfeinert.

Analyse der Istsituation

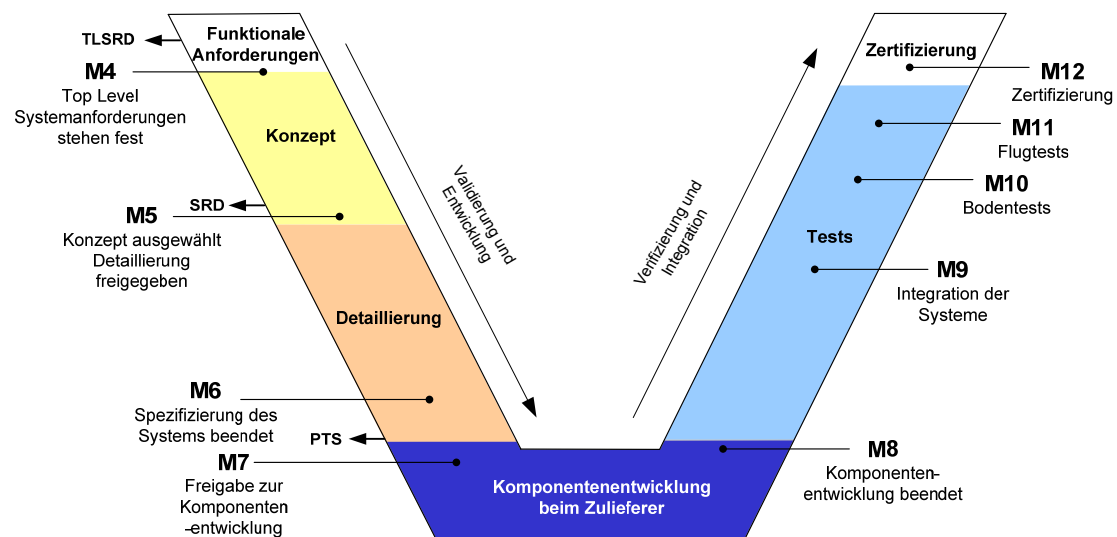


Abbildung 3-2 Entwicklungsphasen im V-Modell [188]

Nachdem auf Flugzeugebene die Machbarkeit überprüft und die Flugzeuganforderungen definiert wurden, werden aus diesen die Hauptsystemanforderungen (Abk. TLSRD; engl.: Top Level System Requirements Document) abgeleitet. In der Konzeptphase erfolgt die Evaluierung der Systemarchitektur; den Abschluss dieser Phase bilden die Systemanforderungsdokumente.

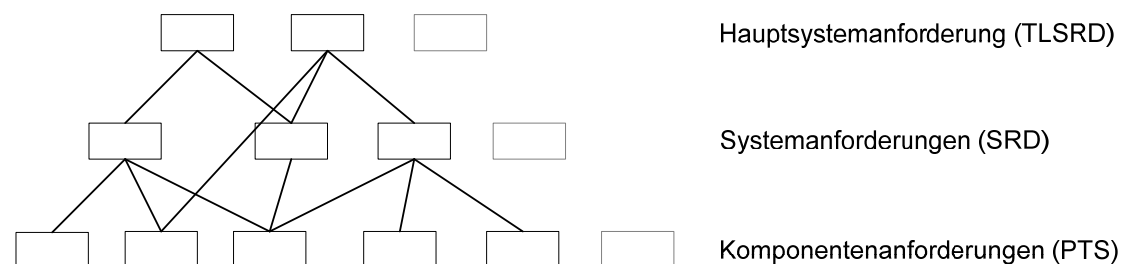


Abbildung 3-3 Top-down-Prinzip der Entwicklungsdokumente

In der folgenden Detaillierungsphase wird das Konzept bis auf Spezifikationslevel verfeinert und definiert. Eine Strategie der Flugzeughersteller ist die Konzentration der Ressourcen auf ihre Kernkompetenzen, sodass maßgebliche Leistungen von den Zulieferern bezogen werden [86]. Das bedeutet, dass die Systeme bis auf Komponentenebene vom Flugzeughersteller spezifiziert werden und die Zulieferer Komponenten entsprechend der Spezifikation (Abk. PTS) entwickeln und zurückliefern. Während der Konzept- und Detaillierungsphasen finden kontinuierlich Integrationstreffen statt, um eventuelle Inkompatibilitäten möglichst frühzeitig erkennen zu können. Gleichzeitig dienen die Treffen zur Validierung des Entwicklungsstands.

Die von Zulieferern entwickelten Komponenten werden von der Testabteilung mithilfe des Spezifikationsdokuments überprüft und anschließend in die Systemumgebung integriert. Nach Abschluss der Boden- und Flugtests erfolgt die Zertifizierung. Wird diese erfolgreich durchlaufen, kann das Flugzeug an den Kunden ausgeliefert werden.

In Abbildung 3-4 ist die parallele Entwicklung der Flugzeugsysteme, in diesem Fall des elektrischen und hydraulischen Systems, dargestellt. Es wird deutlich, dass entlang des gesamten V-Modells Schnittstellen zu den Entwicklungsprozessen anderer Entwicklungsabteilungen bestehen.

Wie bereits in Kapitel 2.3.4 beschrieben existiert derzeit kein Prozess für den Simulationseinsatz in der Entwicklung multidisziplinäre, komplexe Produkte (vgl. [86]). Für den Einsatz von Simulationstechniken in der Flugzeugsystementwicklung werden zwar Empfehlungen ausgesprochen, es fehlen allerdings die verbindlichen Vorschriften. Das führt dazu, dass Simulationen in den verschiedenen an der Entwicklung beteiligten Systemabteilungen sehr unterschiedlich eingesetzt werden, wodurch sich die Wiederverwendbarkeit von Modellen sowie die Analyse der Systeminteraktion erheblich erschweren.

3.1 Istsituation der Systementwicklung

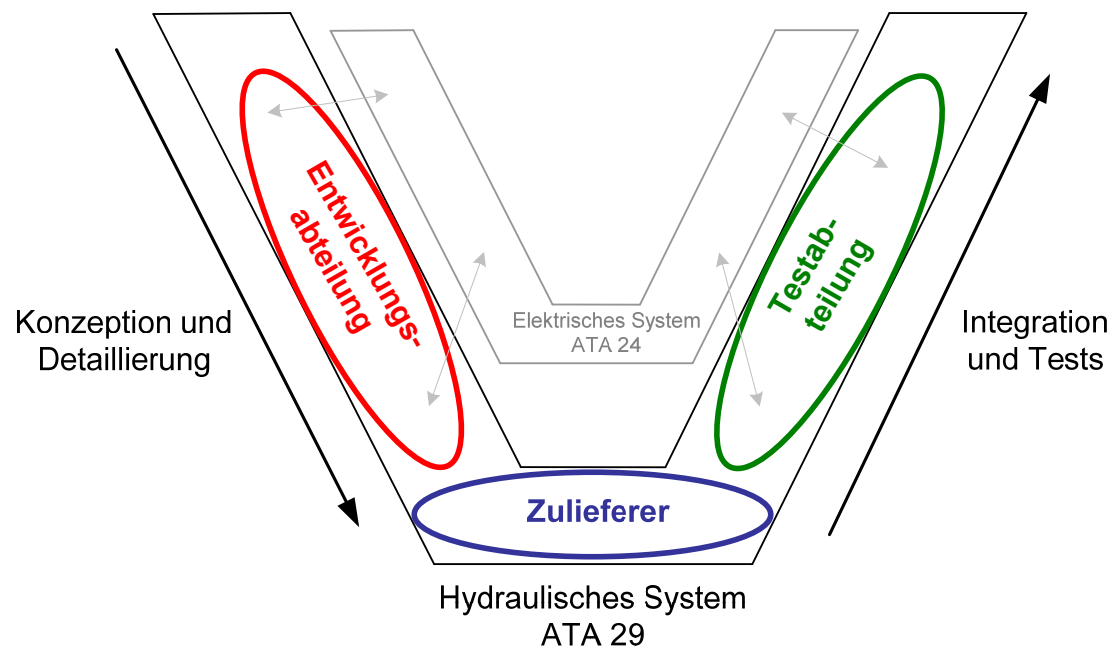


Abbildung 3-4 Systementwicklung mit dem V-Modell und Schnittstelle zu anderen Systemen

Zusätzlich stehen die generierten Simulationsdaten oftmals nur abteilungsintern zur Verfügung, wodurch eine Wissensteilung und damit die Verbesserung des ganzheitlichen Denkens behindert wird.

3.2 Analyse der Systementwicklung in der Flugzeugindustrie

Zur Analyse des bestehenden Prozesses wurden Gespräche mit Entwicklungs- und Testabteilungen in Airbus-Werken in Deutschland und Frankreich durchgeführt. Ziel der Befragungen war es, Informationen zum Einsatz von Simulation und zur allgemeinen Vorgehensweise im Entwicklungsprozess zu erhalten. Hinsichtlich der Simulationstechnik wurde untersucht, welche Simulationsprogramme eingesetzt werden und zu welchem Entwicklungszeitpunkt simuliert wird. Weiterhin wurde untersucht, welche Parameter und Eingangsvariablen für die Simulation in der jeweiligen Abteilung von Interesse sind. Zusätzlich wurde nach den Schnittstellen mit anderen Entwicklungsabteilungen gefragt. Es wurde nach Austauschformaten, CAE-Programmen und Simulationsdatenbanken, die bereits zum Einsatz kommen, gefragt. Schließlich wurden die Experten nach Problemen und Verbesserungsmöglichkeiten, die hinsichtlich des Simulationseinsatzes ihrer Meinung nach bestehen, befragt. Um die derzeitigen Defizite besser identifizieren zu können, wurden die derzeitigen Prozesse der untersuchten Bereiche in ARIS, einem Programm zur Prozesssimulation, modelliert (vgl. Kapitel 7.5). Im Folgenden werden die Untersuchungsergebnisse in die Bereiche Informationsaustausch und Simulationseinsatz gegliedert.

3.2.1 Informationsaustausch

Der Austausch von Informationen und damit das Teilen von Wissen ist für eine erfolgreiche Produktentwicklung von zentraler Bedeutung. In den 90er-Jahren wurden die ersten EDM/PDM-Technologien für die Bereitstellung von Produktinformationen eingeführt (vgl. Kapitel 2.4.2) [173]. Informationen liegen dabei in den unterschiedlichsten Formen (z. B. Dokumente, Geometriedaten, Entwicklungsstatus, Simulationsmodellen und Daten jeglicher Form) vor. Für die Simulation sind in der Entwicklung die Anforderungsdokumente (vgl. Abbildung 3-3), Geometriedaten, Ein- und Ausgangsgrößen der Simulationen sowie ggf. Simulationsmodelle usw. zu speichern und zwischen den Entwicklungsabteilungen auszutauschen.

Zum gegenwärtigen Zeitpunkt werden Daten in einem unternehmensweit zugänglichen Netzlaufwerk abgelegt, dessen Verzeichnisstruktur nicht aktiv verwaltet wird. In einigen Fällen werden projektbezogene Zugriffsrechte der Verzeichnisse vergeben. Dabei ist die Rechtevergabe oft nicht eindeutig geregelt und wird oftmals willkürlich und sporadisch durchgeführt. Hieraus ergibt sich der Nachteil, dass keine automatische Speicherung der Entwick-

Analyse der Istsituation

lung (engl.: history) für den Austausch von Simulationsdaten möglich ist. Ein weiterer Nachteil besteht darin, dass einem Benutzer gestattet ist, direkt auf dem Netzlaufwerk zu arbeiten, sodass beim Bearbeiten einer Datei einem anderen Mitarbeiter der Zugriff auf diese Datei im günstigsten Fall verweigert wird. In ungünstigeren Fällen bearbeiten zwei Benutzer eine Datei zum gleichen Zeitpunkt. Das kann zur Folge haben, dass Änderungen in der Datei unwiderruflich verloren gehen.

Eine weitere Möglichkeit zur Speicherung, Verwaltung und Bereitstellung von Informationen bzw. Daten wird durch den Einsatz von Datenbanken realisiert. Einige der bei Airbus eingesetzten Datenbanken-Systeme werden im Folgenden vorgestellt. Geometriedaten werden über die VPM-Datenbank (engl.: Virtual Product Management) ausgetauscht. In einer speziellen Datenbank werden Zeichnungsdaten für die Fertigung hinterlegt. Anforderungsdokumente werden im EDMS (engl.: Engineering Data Management System) abgelegt. Autorisierte Mitarbeiter können benötigte Daten aus dem EDMS entnehmen. Für die Verwaltung großer Simulationsdatensmengen, wie sie z. B. für die physikalische Simulation typisch sind (vgl. Kapitel 7.2), und zur Visualisierung von 3D-Daten kommt das SimDMS zum Einsatz (vgl. Kapitel 7.4.1). Weiterhin wird bei Airbus eine standortübergreifende Datenbank, die AMDB (engl.: Aircraft Models Database) zum Austausch von Simulationsmodellen verwendet. Die Datenbank wird vom AMO-Team (engl.: Aircraft Models Management) verwaltet. Das AMO-Team ist für die Umsetzung des Airbus-Standards zum Austausch von Simulationsmodellen zuständig. Ziele der AMDB sind:

- Sicherung des Modellaustauschs zwischen den beteiligten Entwicklungsabteilungen
- Organisation, Prüfung und Identifizierung der bestehenden bzw. ausgetauschten Modelle
- Wiederverwendung und Austausch von Integrations- und Validierungserfahrungen
- Verbesserung des Informationsniveaus
- Standardisierung des Modellaustauschs
- Standardisierte Modellkonvertierung in einen plattformunabhängigen Quellcode (z. B. C-Code-Spezifikation)

In der AMDB hinterlegte Daten sind nach ATA-Kapiteln organisiert und können gleichzeitig von mehreren Nutzern verwendet werden. Hauptsächlich wird die AMDB von standortübergreifenden Entwicklungsabteilungen zum Austausch von Modellen eingesetzt. Für die problemlose Wiederverwendung der Modelle ist es erforderlich, dass diese mit einer Schnittstelle in C oder Fortran-Code bereitgestellt werden. Um Modelle in der AMDB ablegen zu können, müssen zudem eine Reihe von modellbegleitenden Dokumenten erstellt werden. Diese Anforderungen an die Modellablage bedeuten einen erhöhten Mehraufwand, sodass ein Großteil der in den verschiedenen Entwicklungsabteilungen erstellten Modelle nicht konform mit den Unternehmensrichtlinien modelliert wird. Diese Modelle werden dann wieder abteilungsintern lokal gespeichert, sodass sie für Dritte nicht sichtbar sind. Aufgrund der fehlenden Dokumente sind diese Modelle innerhalb kürzester Zeit selbst für den Ersteller nur sehr schwer lesbar, sodass es zu einem Mehraufwand bei der Wiederverwendung kommt. Zudem ist das Auffinden der Modelle durch die uneinheitliche Speicherung erheblich erschwert. Für den Fall des Ausscheidens eines Mitarbeiters kann so Wissen unwiederbringlich verloren gehen. Der Einsatz von Simulation und der Austausch der Modelle und Ergebnisse müssen von den beteiligten Personen gelebt werden. Hierfür ist es wichtig, dass durch den Modellaustausch ein Nutzen für alle Beteiligten gewährleistet wird.

Die im Vorangegangenen beschriebenen Datenbanksysteme stellen nur einen marginalen Auszug der zurzeit eingesetzten Systeme dar. Es existiert eine Vielzahl an unabhängigen, nicht vernetzten Datenbanksystemen, die über individuelle, nicht synchronisierte Zugangsberechtigungen verfügen. Jeder Benutzer muss sich mühsam für die benötigten Datenbanken bei verschiedenen Administratoren freischalten lassen. Für eine Vereinfachung der Informationsbeschaffung, die Optimierung der Kommunikation und eine erhöhte Wissensteilung im Unternehmen ist die Vernetzung der Informationsbereitstellung von großer Bedeutung.

Zusätzlich zu den bereits vorgestellten Möglichkeiten des Datenaustauschs kommen weitere Verfahren zum Einsatz. Seit Mitte der 90er-Jahre des vergangenen Jahrhunderts hat sich die elektronische Post, kurz E-Mail, zu einem der Hauptkommunikationsmittel des privaten wie auch des beruflichen Lebens entwickelt. Die E-Mail eignet sich auch zum Austausch von Daten, allerdings wird immer ein direkter Kontakt benötigt, d. h. die am Austausch

3.2 Analyse der Systementwicklung in der Flugzeugindustrie

beteiligten Personen müssen sich zumindest unidirektional kennen. Mit anderen Worten bedeutet das, dass die Informationsquelle, in diesem Fall ein E-Mail-Kontakt, bekannt sein muss, welche die benötigten Informationen liefern kann. Neben der Tatsache, dass die Informationsbeschaffung auf diesem Weg immer indirekt ist und somit eine Verzögerung der Informationsbereitstellung darstellt, muss der Nutzer über ein Personennetzwerk verfügen. So gelangen Nutzer, die seit Langem in einem Unternehmen beschäftigt sind und über umfangreiche Kontakte verfügen, schneller und einfacher an Informationen als Personen, die erst seit Kurzem im Unternehmen beschäftigt sind. Ein weiteres Vorgehen zum Austausch von Daten, das in der Praxis häufig verwendet wird, ist der Datenaustausch per mobilem Speichermedium. Nachteile dieser Methode sind der lokal beschränkte Einsatz sowie Sicherheitsrisiken bei Verlust. Dass es sich beim Austausch von Daten per E-Mail oder mobilen Speichermedium um kein allgemeingültiges und unternehmensweit einsetzbares Verfahren handelt, ist offensichtlich.

3.2.2 Simulationseinsatz in der Entwicklung von Flugzeugsystemen

In den letzten Jahren konnte ein stetiger Anstieg der Systemkomplexität sowie der Systeminteraktionen festgestellt werden. Dieser Trend wird sich zukünftig noch verstärken. Ursachen hierfür sind disziplinübergreifende Systementwicklungen, die Globalisierung, die Marktsituation, der Variantenreichtum und vieles mehr [86]. In Analogie zur Produktkomplexität ist die Leistungsfähigkeit der Computer ebenfalls angestiegen. Um der steigenden Komplexität gerecht zu werden, ist der logische nächste Schritt die Integration und Kopplung von Computerprogrammen in der Flugzeugentwicklung [116].

Zum gegenwärtigen Zeitpunkt werden bereits unterschiedlichste CAE-Programme in der Praxis eingesetzt. Von einem durchgängigen Einsatz von Simulation, einer Wiederverwendung von Modellen sowie einer effizienten Wissensteilung ist man in der Flugzeugentwicklung allerdings noch weit entfernt. Im Folgenden wird der derzeitige Einsatz von Simulationswerkzeugen in der Flugzeugentwicklung beschrieben. Dafür wurde das Vorgehen verschiedener Entwicklungsabteilungen mit dem Programm ARIS modelliert. Ein Beispiel für einen in ARIS modellierten Prozess ist im Anhang dargestellt (siehe Kapitel 7.5).

In den vorangegangenen Kapiteln wurde deutlich, dass der Grundstein einer erfolgreichen Produktentwicklung in den frühen Entwicklungsphasen liegt [86], [55]. Die Fehlerkorrekturkosten steigen während der Entwicklung kontinuierlich an [114], aus diesem Grund ist die frühzeitig Überprüfung der Produktentwicklung von besonderer Wichtigkeit.

In der Praxis zeigt sich ein gegenteiliges Bild: In der frühen Systementwicklung werden nur sehr wenige Ressourcen eingesetzt. Diese reichen nicht aus, um den Einsatz von Simulationsprogrammen in dieser frühen Phase zu gewährleisten. Die Auswahl eines Konzepts ist empirisch basiert, d. h., die an der Entwicklung Beteiligten greifen auf individuelles sowie im Unternehmen vorhandenes Wissen zurück. Eine Analyse von Konzeptalternativen findet kaum oder gar nicht statt. Wenn überhaupt, so werden einfache Berechnungsmethoden eingesetzt. Damit wird zwar immer eine Lösung bzw. ein Konzept ausgewählt, ob es sich dabei um das am besten geeignete Konzept zum gegenwärtigen Zeitpunkt handelt, kann jedoch nicht festgestellt werden. Dieses Problem wird bereits von Vajna beschrieben. Oftmals wird aus Gründen der Arbeitsbelastung nur eine Lösung betrachtet. Der Entwickler entscheidet sich für die nächstbeste auf seinen Erfahrungen basierende Lösung. Vajna beschreibt einen Lösungsraum, in dem sämtliche Lösungen enthalten sind. Die Identifikation der optimalen Lösung für multidimensionale Anforderungen kann dabei durch den Einsatz von Computern erheblich vereinfacht werden (z. B. durch den Einsatz von Simulationsprogrammen) [174], [175].

Abbildung 3-5 zeigt den Einsatz von Simulation während der Systementwicklung. Es wird deutlich, dass Simulation in der späten Konzeptphase sowie zur Detaillierung und Spezifizierung des Systems eingesetzt wird. Je nach Vertrag werden nach der Komponentenentwicklung von den Zulieferern zusätzlich zu den Hardwarekomponenten noch Komponentenmodelle geliefert. Die Testabteilung setzt Simulationsmodelle z. B. im Rahmen von Hardware in the Loop-Tests ein. Im Folgenden werden die Konzept-, Detaillierungs-, Entwicklungs- sowie die Testphasen detailliert beschrieben.

Analyse der Istsituation

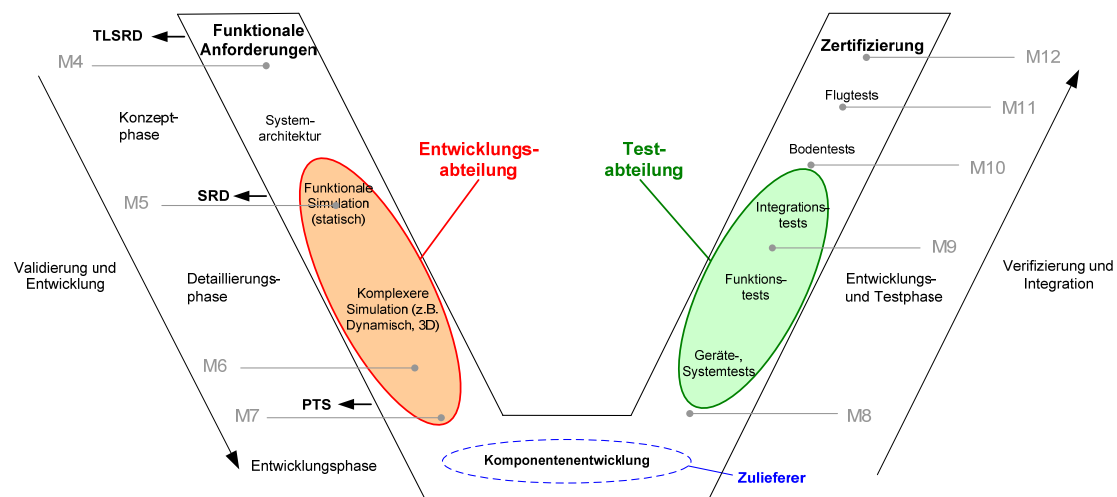


Abbildung 3-5 Allgemeiner Einsatz von Simulation in der Flugzeugsystementwicklung

Konzeptphase (M4–M5)

In der Konzeptphase wird die Systemarchitektur basierend auf den Hauptsystemanforderungen (vgl. Seite 27), entwickelt. In der späten Konzeptentwicklung findet eine Überprüfung der funktionalen Systemanforderungen mithilfe von Simulationsprogrammen statt. Der Fokus der funktionalen Simulation liegt in der Modellierung des Gesamtsystems mit allen Hauptkomponenten. Die Modellierung erfolgt auf einer hohen Abstraktionsebene, sodass eine statische Funktionsanalyse der Hauptkomponenten und der Interaktionen zwischen ihnen möglich ist. Typischerweise kommen hierfür multidisziplinäre Simulationsprogramme zum Einsatz. Bedingt durch das Top-down-Prinzip bestehen in der frühen Entwicklungsphase zahlreiche Schnittstellen zu anderen Systemen. Für die Simulationsdurchführung werden viele Eingangsgrößen anderer Systeme benötigt, z. B. werden Informationen der jeweiligen Eingangsgrößen benötigt, welche aus anderen Abteilungen bezogen werden müssen. Diese Tatsache führt zu dem bereits angesprochenen Problem der Informationsbereitstellung (vgl. Kapitel 2.4). Wer Informationen benötigt, muss sich diese beschaffen, er ist damit in der Holpflicht. Für den Entwickler bedeutet dies einen hohen Informationsbeschaffungsaufwand. Den Abschluss dieser Phase bilden die Systemarchitektur, die in der weiteren Entwicklung detailliert und spezifiziert wird, und das Systemanforderungsdokument.

Auffällig ist, dass die frühe Konzeptphase nicht durch Simulation unterstützt wird und somit nur die nächstbeste Systemarchitektur für die Weiterentwicklung zur Verfügung steht. Es wird sich damit in einer für die Systementwicklung entscheidenden Phase auf ein Konzept festgelegt, das keinem Optimierungsprozess unterzogen wurde. Stattdessen erfolgt die Auswahl einer Systemarchitektur ausschließlich auf Basis von Erfahrungswerten aus früheren Entwicklungsprojekten. Dieses Vorgehen hat zur Folge, dass vor allem auf altbewährte Konzepte zurückgegriffen wird, die lediglich an die aktuellen Rahmenbedingungen angepasst werden. Der Innovationsgrad in der Konzeptentwicklung ist durch dieses Vorgehen stark eingeschränkt. Die Erfüllung der Anforderungen wird mit einfachen Berechnungsmethoden überprüft, die keine Analysen multidimensionaler Kriterien erlauben. Änderungen an der Systemarchitektur nach Abschluss der Konzeptphase kommen nur dann vor, wenn die Funktionalität nicht gewährleistet werden kann.

„In der Folge gibt es in komplexen Systemen eine gewisse Tendenz zur Übersteuerung und zu Reparaturverhalten. Das heißt, lediglich akute Missstände werden beseitigt, ohne die dahinterliegenden Ursachen-Wirkungs-Zusammenhänge zu berücksichtigen [86].“

Definitionsphase (M5–M7)

Mit fortschreitender Entwicklungszeit steigt der Detaillierungsgrad der Modelle. Analog zur Systemdekomposition in Subsysteme und Komponenten werden Submodelle und Komponentenmodelle erstellt. Die Simulation wird in dieser Phase zur Analyse des dynamischen (transienten) Systemverhaltens eingesetzt. Im Zuge der voranschreitenden Entwicklung kommt es zu sogenannten Insellösungen, alleinstehenden Simulationen, die zur Validierung und Spezifizierung einzelner Anforderungen eingesetzt werden. Ein Nachteil

3.2 Analyse der Systementwicklung in der Flugzeugindustrie

dieses Vorgehens ist die Vernachlässigung der Gesamtsysteminteraktionen, allerdings ist die Modellierung unkompliziert, da keine Schnittstellen zu angrenzenden Systemen berücksichtigt werden müssen. In der späten Detaillierungsphase sind Geometriedaten vorhanden, sodass zusätzlich zu der Systemsimulation auch gestaltorientierte Simulationsprogramme eingesetzt werden können. In Einzelfällen dienen bereits die Ergebnisse der Systemsimulation als Eingangsparameter bzw. -größen für die gestaltorientierte Simulation. Eine Lösung des hierfür benötigten Informationsaustausches wird in Kapitel 5.3.1 vorgestellt. Oftmals werden fehlende Simulationsparameter durch Erfahrungswerte ersetzt; stehen zu einem späteren Zeitpunkt die realen Parameter zur Verfügung, wird die Simulation erneut durchgeführt. Dies führt besonders im Fall der gestaltorientierten Simulationen zu einem hohen Mehraufwand. Die Ziele des Simulationseinsatzes in der Detaillierungsphase sind:

- Validierung der Anforderungen
- Verbesserung des Systemverständnisses
- Aussagen über technische Auswirkungen auf Designfragen
- Unterstützung der Spezifikationserstellung
- Reduzierung der Flug- und Bodentests
- Aussagen über Systeminteraktionen
- Transiente Systemanalyse
- Frühzeitige Identifikation von Entwicklungsfehlern

Hierbei werden Systeminteraktionen im derzeitigen Prozess nur in den wenigsten Fällen mithilfe von Simulationstechniken analysiert. Die in der Konzept- und Detaillierungsphase entwickelten Modelle werden entweder lokal gespeichert oder in der AMDB abgelegt. Wie bereits beschrieben (vgl. Kapitel 3.2.1) müssen für das Ablegen in der AMDB die Unternehmensrichtlinien eingehalten werden. Dies bedeutet einen zusätzlichen Arbeitsaufwand, der in den seltensten Fällen aufgebracht wird, sodass ein Großteil der erstellten Modelle lokal gespeichert wird und somit nicht über ein Informationsmanagementsystem konzernweit verfügbar ist. Hinzu kommt, dass Entwicklungsabteilungen durch die Freigabe ihrer Modelle einen Wissensverlust fürchten.

Den Abschluss der Detaillierungsphase bildet die endgültige Spezifikation des Systems, die sogenannte Purchaser Technical Specification (Abk. PTS). Sie bildet die Vertragsgrundlage mit den Zulieferern.

Zusammenfassend lässt sich sagen, dass es sich bei der Entwicklung von Flugzeugsystemen um einen systemübergreifenden Prozess handelt, der durch Systemwechselwirkungen gekennzeichnet ist. Die Überprüfung dieser Wechselwirkungen hinsichtlich der Erfüllung von Funktionen und der Kompatibilität von Schnittstellen findet während der Detaillierungsphase nicht statt. Eine Überprüfung der Systeminteraktionen wird erst im Rahmen der Integrationstests in der Testabteilung durchgeführt. Zudem werden lediglich die Fragestellungen des Simulationsauftraggebers beantwortet, ein Großteil der Ergebnisse und Informationen aus den durchgeführten Simulationen bleibt unverwertet in der Fachabteilung zurück. Bedingt durch die geringe systemübergreifende Wissensteilung in der Entwicklung besteht damit die Gefahr der Entwicklung von Insellösungen. Inkompatibilitäten, die erst während der Integrationstests der Systeme erkannt werden, führen zu aufwendigen und teuren Änderungsschleifen im Entwicklungsprozess. Es sei an dieser Stelle darauf hingewiesen, dass zwar systemübergreifende Integrationstreffen durchgeführt werden, diese allerdings im Wesentlichen nur geometrische Unstimmigkeiten zwischen den Abteilungen klären. Ein weiteres Problem besteht darin, dass gleiche Modelle von verschiedenen Abteilungen mehrfach erstellt werden. Dieser Mehraufwand könnte durch ein geeignetes Informationsmanagement erheblich reduziert werden.

Komponentenentwicklung (M7–M8)

Nachdem die Auftragsvergabe erfolgt ist, werden die Systemkomponenten vom Zulieferer erstellt. In der Regel wird in der Vertragsvereinbarung mit dem Zulieferer festgelegt, dass neben der Hardware auch Simulationsmodelle der Komponente zu liefern sind, die in die anschließende Testphase eingebunden werden können. Die Integration und Verifizierung auf Geräteebene wird gewöhnlich vom Zulieferer durchgeführt.

Testphase (M8–M10)

Die Testphase befindet sich im rechten Strang des V-Modells (vgl. Abbildung 2-1) in dem die Einzelkomponenten zu Systemen und diese letztendlich zum Produkt Flugzeug integriert werden. In den einzelnen Testphasen wird die Hardware anhand der Anforderungen verifiziert. Für den Fall, dass Entwicklungsfehler identifiziert werden, muss der Entwicklungsprozess ab der jeweils korrespondierenden Phase im linken Strang des V-Modells erneut durchlaufen werden.

Die Testabteilung beginnt mit dem Aufbau der Testumgebung erst, nachdem die Lieferung der Geräte bzw. Komponenten durch die Hersteller erfolgt ist. Die Testumgebung besteht zum einen aus dem Teststand mit den erforderlichen Schnittstellen zur Aufnahme der Hardwarekomponenten und zum anderen aus einer Softwareschnittstelle, die zur Einbindung der Komponenten- und der Umgebungsmodelle dient. Das Umgebungsmodell simuliert die Systemumgebung (z. B. Temperatur, Druck usw.) und die Eingangsparameter angrenzender Systeme. Die Softwareschnittstelle der Testumgebung ermöglicht Interaktionen zwischen Hardware und Softwaremodellen (vgl. Abbildung 3-6).

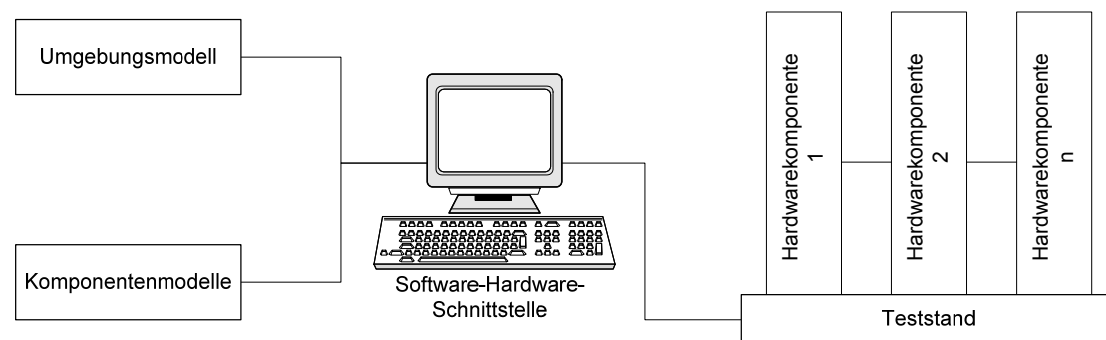


Abbildung 3-6 Aufbau der Testumgebung

Die HIL-Simulation ermöglicht z. B., fehlende Hardwarekomponenten durch Simulationsmodelle zu ersetzen und diese im Verbund mit Hardwarekomponenten zu analysieren. Voraussetzung hierfür ist die Echtzeitfähigkeit der eingesetzten Modelle. Die Systemumgebung wird in einem Umgebungsmodell simuliert, so wird es möglich, das System unter verschiedenen Umgebungsbedingungen zu testen, ohne diese real zu erzeugen (z. B. Windkanal, Kältekammer usw.). Kostspielige Tests unter realen Bedingungen werden durch diese Maßnahme auf ein Minimum reduziert. Unter ökonomischen Gesichtspunkten betrachtet, gewährleistet eine HIL-Simulation die gefahrlose Analyse kritischer Szenarien (z. B. Überlastfälle, Notfallszenarien usw.), die zur Zerstörung oder Beschädigung von Komponenten oder des gesamten Systems führen können. Zurzeit werden keine globalen Umgebungsmodelle verwendet. Das bedeutet, dass die Umgebungsmodelle, die bereits in der Konzept- und Detaillierungsphase für die Simulation erstellt wurden, in der späteren Testphase nicht wieder verwendet werden. Die Testabteilung entwickelt eigene Umgebungsmodelle, dieser Mehraufwand könnte durch den Einsatz globaler Umgebungsmodelle, die für alle Entwicklungsphasen Gültigkeit besitzen, reduziert werden. Zwischen den Entwicklungsabteilungen und den Testabteilungen besteht hinsichtlich der Modelle kaum Austausch. Ein Grund hierfür liegt darin, dass die Modelle aus der Entwicklungsabteilung das Systemverhalten repräsentieren, in der Testabteilung jedoch in erster Linie die Schnittstellen zwischen den Systemen bzw. Komponenten überprüft werden. Allerdings könnten die Testabteilungen Modelle der Entwicklungsabteilung nutzen, um parallel zur Komponentenentwicklung frühzeitig mit der Erstellung der Testvorrichtung zu beginnen. Zurzeit werden nur echtzeitfähige Regelstrecken der funktionalen Modelle, die der AMDB entnommen werden, in die Testumgebung integriert (vgl. Abbildung 3-7).

In der Testphase werden die Geräte-, Funktions- und Integrationstests sequenziell durchlaufen. Zunächst werden die Geräte bzw. Systeme auf die Erfüllung der Spezifikation hin überprüft. In den anschließenden Funktionstests wird eine Funktion (z. B. Bereitstellen einer bestimmten Temperatur) kontrolliert.

3.2 Analyse der Systementwicklung in der Flugzeugindustrie

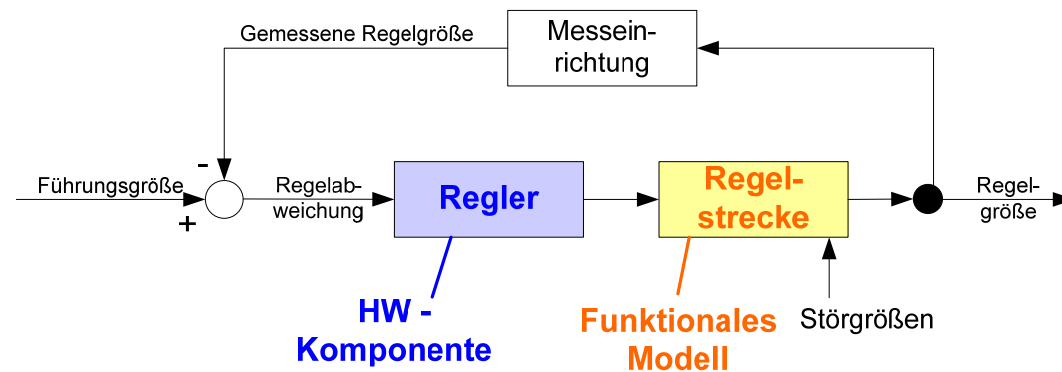


Abbildung 3-7 Integration funktionaler Modelle in die Testumgebung

Es folgt die Überprüfung der Systemschnittstellen während der Integrationstests. Für den Fall, dass in den Phasen nach den erfolgreichen Integrationstests keine Probleme auftreten, ist die virtuelle Begleitung der Systementwicklung mit Erreichen des zwölften Meilensteins (Flugzeugmontage beendet; vgl. Abbildung 3-1) beendet. In den anschließenden Boden- und Flugtests sind alle Systemkomponenten in Form von Hardware an Bord des Flugzeugs integriert. Zudem sind aus Zertifizierungsgründen diese beiden Tests mit der realen Hardware zu absolvieren. Eine Zertifizierung anhand von Simulationsuntersuchungen ist zurzeit noch nicht möglich.

3.3 Zusammenfassung der Defizite im bestehenden Entwicklungsprozess

In diesem Kapitel werden die in den zwei vorangegangenen Abschnitten beschriebenen Defizite zusammengefasst und durch bekannte Defizite aus der Literatur ergänzt. Basierend auf diesen Defiziten werden im folgenden Kapitel die Anforderungen an ein Konzept für den durchgängigen Einsatz von Simulationstechniken in der Entwicklung von Flugzeugsystemen abgeleitet.

Die Systeme und ihre Interaktionen werden immer komplexer. Ursachen dafür sind der harte Konkurrenzkampf, die Übersättigung der Märkte, der Variantenreichtum und die zunehmende Zahl an multidisziplinären Systemen [86]. Seit Einführung der Mechatronik 1969 stieg die Zahl der fachdisziplinübergreifenden Systementwicklungen rapide an [86]. Hinter dem Begriff der Mechatronik verbirgt sich das enge synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und bei der Herstellung industrieller Erzeugnisse sowie bei der Prozessgestaltung [179]. „Bislang galt die Elektrotechnik - bzw. Softwareentwicklung - als Dienstleister für die Maschinenbaubranche“ [86]. Für zukünftige Produkte, die sich durch Interaktionen heterogener physikalischer Bereiche auszeichnen, muss ein angepasster Produktentstehungsprozess entwickelt werden. Gleiches gilt auch für den Systementwicklungsprozess im Flugzeugbau. Dieser beginnt mit der Festlegung der Hauptsystemanforderungen und endet mit der Zertifizierung (M4–M12; vgl. Abbildung 3-1). Der derzeitige Flugzeugsystementwicklungsprozess zeichnet sich im Wesentlichen durch die vier folgenden Defizite aus:

Defizit 1: kein durchgängiger Simulationseinsatz im Prozess

Zum gegenwärtigen Zeitpunkt werden Simulationstechniken nicht in allen Phasen der Flugzeugsystementwicklung eingesetzt. So werden z. B. in der frühen Konzeptphase verschiedene Systemarchitekturen nicht virtuell analysiert. Die Festlegung auf ein Konzept erfolgt empirisch, es wird somit nur die nächstbeste Lösung gefunden. Wie bereits in Kapitel 2.3 erwähnt (vgl. Abbildung 2-6), liegt der Grundstein für einen erfolgreichen Entwicklungsprozess in den frühen Phasen [86], da der Einfluss auf die Entwicklung der Gesamtkosten hier am größten ist [114], sodass gerade zu Beginn der Entwicklung die bestmögliche Lösung gefunden werden muss.

Durch das frühe Festlegen auf eine Systemarchitektur kommt es in der Folge zum Reparaturverhalten in der Entwicklung. Es werden lediglich die akuten Probleme beseitigt, die Ursachen, die bereits in einer fehlerhaften Konzeptauswahl begründet sein können, werden nicht beseitigt. Ein weiteres Problem in der frühen Entwicklungsphase besteht darin, dass die

Analyse der Istsituation

Konzeptauswahl auf Basis vorangegangener Projekte getroffen wird. Es wird auf Altbewährtes gesetzt, wodurch der Innovationsgrad des Produkts eingeschränkt wird.

In den Unternehmensdokumenten existiert keine Richtlinie, die den Einsatz von Simulationstechniken vorschreibt und regelt. Diese Tatsache wurde bereits von Franke erkannt, der schreibt, dass es unter anderem an Assistenzsystemen und Vorgehensrichtlinien, die den Entwicklungsprozess system- und phasenübergreifend begleiten und die Vermeidung von Wiederholungsfehlern unterstützen, fehlt [86].

Defizit 2: keine Berücksichtigung der Systeminteraktionen

Die Entwicklung von Flugzeugsystemen zeichnet sich durch Interaktionen komplexer integrierter Systeme aus. Für eine erfolgreiche Entwicklung ist es von großer Bedeutung, die Wechselwirkungen der verschiedenen Fachdisziplinen zu berücksichtigen. Im derzeitigen Entwicklungsprozess von Flugzeugsystemen werden diese allerdings kaum berücksichtigt.

Zurzeit verwenden die Fachabteilungen unterschiedliche Berechnungsmethoden und Simulationstechniken, deren Ergebnisse aus Angst vor Wissensverlust nur lokal in der jeweiligen Abteilung zur Verfügung stehen. Ein weiteres Problem besteht darin, dass die an der multidisziplinären Systementwicklung beteiligten Abteilungen eigene Begriffswelten, Methoden, Verfahren und Werkzeuge aufweisen, wodurch es zu inkonsistenten Teilentwicklungen und nicht berücksichtigten Wechselwirkungen kommt [86]. Hierdurch bedingt werden Systeminkompatibilitäten erst während der Integrationstests erkannt. Dies führt zu zeit- und kostenintensiven Rücksprüngen im Entwicklungsprozess.

Die Ursache für ein späteres Fehlverhalten von Systemen liegt oftmals in der Vernachlässigung der gegenseitigen Beeinflussungen der Systemkomponenten und externer Einflussparameter. Zudem sind oftmals keine Schnittstellen zwischen den Komponenten definiert [86]. So wurden bei der Entwicklung der Verkabelung des elektrischen Systems des A380 zwei unterschiedliche CATIA-Softwareversionen (engl.: Computer Aided Three-Dimensional Interactive Application) eingesetzt. Diese fehlende Abstimmung der Schnittstellen führte zu erheblichen Verzögerungen und Mehrkosten in der Entwicklung.

Defizit 3: Prozessverzögerungen durch fehlende Zusammenarbeit

Ein bereits zum gegenwärtigen Zeitpunkt vermeidbares Problem besteht in der fehlenden Zusammenarbeit zwischen den in die Entwicklung involvierten Fachabteilungen. Wie im Vorangegangenen erwähnt, besteht die Angst des Wissensverlusts durch den Austausch von Informationen, Daten, Modellen und anderen Entwicklungsergebnissen. Durch eine Intensivierung des Wissensaustausches könnten abteilungsübergreifende Inkompatibilitäten allerdings frühzeitig erkannt sowie das Verständnis für das Gesamtsystem erhöht werden. Die verstärkte Zusammenarbeit würde eine Gesamtsystemoptimierung gewährleisten und nicht wie bisher nur eine Optimierung der Einzelkomponenten (Insellösungen).

Die bereits vorhandenen Schnittstellen für den Modellaustausch werden kaum genutzt, da sie für den Einzelnen zunächst einen Mehraufwand bedeuten. Es wird nicht erkannt, dass dieser Mehraufwand einmalig ist und sich innerhalb kürzester Zeit durch die Integration von Fremdmodellen im eigenen System amortisieren würde. Als Resultat des mangelhaften Modellaustauschs kommt es immer wieder zu redundanten Entwicklungen bereits existierender Modelle.

Die Zusammenarbeit von Experten verschiedener Fachdisziplinen stellt zusätzliche Anforderungen an deren Qualifikation. Neben Fachwissen werden Teamfähigkeit und interdisziplinäre Kommunikationskompetenz benötigt [86]. Durch den Einsatz einer standardisierten Systembeschreibungssprache könnten die verschiedenen Begriffswelten der einzelnen Fachdisziplinen vereinheitlicht werden. Eine Anforderung an die Qualifikation der Experten bleibt damit bestehen: das Erlernen einer neuen multidisziplinären Systembeschreibungssprache.

Ein weiteres Defizit besteht darin, dass die Testvorrichtung erst nach der Komponentenentwicklung, also nach Lieferung der Hardware, erstellt werden kann. Die für die Tests benötigten Simulationsmodelle, das heißt das Umgebungsmodell sowie die Komponentenmodelle der fehlenden Hardwaresystemkomponenten, werden von der Testabteilung eigenständig erstellt. Insgesamt entsteht durch den späten Beginn der Erstellung der Vorrichtung, den geringen Modellaustausch zwischen Fach- und Testabteilung sowie den großen Eigenanteil bei der Erstellung der Modelle ein erheblicher Mehraufwand und damit eine Verlängerung der Entwicklungszeit.

3.3 Zusammenfassung der Defizite im bestehenden Entwicklungsprozess

Defizit 4: unzureichender Wissens- bzw. Informationsaustausch

Komplexe Unternehmensstrukturen mit mehreren Entwicklungsstandorten sowie Teilentwicklungen bei Zulieferern, wie sie in der Flugzeugindustrie vorherrschen, erschweren den Wissensaustausch zwischen den an der Entwicklung beteiligten Fachabteilungen. Inkonsistente Datenquellen und inkompatible Datenverarbeitungssysteme intensivieren dieses Problem zusätzlich. Zum gegenwärtigen Zeitpunkt kommen unterschiedliche Datenbanken und Datenmanagementsysteme zum Einsatz. Eine Synchronisation dieser Datenquellen ist nicht vorhanden, d. h., der Benutzer besitzt verschiedene Zugriffsrechte und muss unterschiedliche Datenquellen benutzen, um an die benötigten Informationen zu gelangen. Durch eine Vielzahl von Passwörtern, unterschiedliche grafische Benutzeroberflächen sowie unterschiedliche Datenverwaltungsstrukturen ist das Auffinden der benötigten Informationen mit einem erheblichen Aufwand verbunden. Oftmals ist das Neuentwickeln einfacher Simulationsmodelle schneller als das Finden bereits verfügbarer validierter Modelle, und so wird der Entwickler verleitet, neue Modelle zu erstellen. Es kommt zu Doppelarbeiten, und zusätzlich existieren jetzt mehrere gleiche Modelle, die wiederum zu Verwirrungen führen und die Dateiverwaltung unnötig belasten. Häufig werden Informationen nur lokal, nicht aber global zur Verfügung gestellt, sie stehen also nicht unternehmensweit zur Verfügung. Anstelle von aktuellen Informationen wird daher häufig auf Erfahrungswerte zurückgegriffen. Eine Folge hiervon ist die unkoordinierte autarke Systementwicklung in den einzelnen Entwicklungsabteilungen. Oftmals ist der Status einer Information (Version, Gültigkeit usw.) unbekannt. Dies hat zur Folge, dass veraltete oder nicht validierte Modelle in der Simulation verwendet werden. In Tabelle 3-1 sind alle Defizite aus diesem und den vorangegangenen Kapiteln zusammengefasst.

Im Rahmen der Analyse wurde deutlich, dass der bestehende Systementwicklungsprozess eine Reihe von Defiziten aufweist. Die derzeitigen Unternehmensrichtlinien in der Luftfahrtindustrie schreiben kein standardisiertes Vorgehen für den Einsatz von Simulation im Entwicklungsprozess vor. Es konnte gezeigt werden, dass die gegenwärtige Entwicklungsmethodik die Besonderheiten der systemübergreifenden und komplexen Flugzeugentwicklung unzureichend berücksichtigt. Aus diesem Grund ist die Konzeption eines integrierten Prozessmodells erforderlich, welches die Simulationstechnik optimal in den Prozess einbezieht und die entsprechenden Voraussetzungen, wie eine transparente Datenbereitstellung, schafft.

Analyse der Istsituation

Defizit 1 kein durchgängiger Simulationseinsatz im Prozess	
	Keine Unterstützung der Konzeptfindung durch Simulation
	Frühe Festlegung auf eine Systemarchitektur
	Geringer Innovationsgrad in der Konzeptentwicklung
	Fehlerreparatur anstelle von Ursachenbehebung
	Empirisch basierte Auswahl des nächstbesten Konzepts
	Kein standardisiertes Vorgehen für den Einsatz von Simulation während der Entwicklung
	Ungeregelte Weiterverwendung von Modellen
Defizit 2: keine Berücksichtigung der Systeminteraktionen	
	Systemübergreifende Überprüfung der Entwicklung erst während der Integrationstests
	Fehlende Überprüfung der Wechselwirkungen
	Aufwendige Änderungsschleifen bei Feststellen von Inkompatibilitäten
	Häufige Iterationen aufgrund von neuen Parametern anderer Systeme
	Keine Weitergabe von Simulationsergebnissen an angrenzende Fachabteilungen
	Fehlende Schnittstellenbereitstellung
Defizit 3: Prozessverzögerungen durch fehlende Zusammenarbeit	
	Späte Erstellung und Verifizierung der Testvorrichtungen
	Mehraufwand durch die erneute Entwicklung bereits vorhandener Modelle
	Eingeschränkter Informationsaustausch aufgrund von Angst vor Wissensverlust
	Fehlen einer einheitlichen Systembeschreibungssprache
Defizit 4: unzureichender Wissens- bzw. Informationsaustausch	
	Verwendung von Annahmen für Simulationsparameter
	Keine einheitliche Bereitstellung von Informationen
	Informationsverluste durch kaum vorhandene Informationsrückläufe
	Unklarer Status der Informationsqualität
	Inkonsistenz in der Daten- und Modellverwendung
	Keine standardisierte Klassifizierung von Modellen und Daten
	Ungeregelter Datenaustausch
	Viele und uneinheitliche Datenbanken, -managementsysteme
	Viele verschiedene Zugriffsrechte auf bestehende Datenbanken
	Unzureichender Datenaustausch auf Basis von persönlichen Absprachen
	Holpflicht in Bezug auf Informationen für den Entwickler
	Kaum systemübergreifender Modellaustausch
	Abteilungsinterne (lokale) Ablage von Modellierungsinformationen
	Unterschiedliche Datenverwaltungsstrukturen und Benutzeroberflächen

Tabelle 3-1 Defizite des derzeitigen Entwicklungsprozesses von Flugzeugsystemen

Basierend auf den identifizierten Defiziten werden im folgenden Kapitel die Anforderungen an ein solches Prozessmodell aufgestellt.

3.4 Anforderungen

In Analogie zur Entwicklung mithilfe des V-Modells müssen für die Entwicklung eines Konzeptes zur durchgängigen Simulationsunterstützung zunächst die Anforderungen definiert werden. Dabei ist darauf zu achten, dass die Anforderungen keine Lösung beschreiben. Dies würde die Möglichkeit verbauen, alternative Lösungen zu evaluieren [186]. Im folgenden Kapitel 4 werden Konzeptalternativen verglichen und das Konzept, welches die Anforderungen am besten erfüllt, ausgewählt. Zu diesem Zweck erfolgt zunächst eine Definition der Hauptanforderungen, die in einem weiteren Schritt detailliert werden.

Die Anforderungen an das Konzept ergeben sich direkt aus den im vorangegangenen Kapitel identifizierten Defiziten. Das größte Problem im derzeitigen Entwicklungsprozess stellt die fehlende Koordination und Vernetzung zwischen den Entwicklungsabteilungen dar. Die Folgen sind Missverständnisse, Verzögerungen, Inkompatibilitäten, redundante Entwicklungsaktivitäten und damit hohe Entwicklungskosten. Die Hauptanforderungen an ein zukünftiges Konzept lassen sich in die drei Bereiche Modellierung, Einsatz von Simulationstechniken im Prozess und Informationsmanagement einteilen (vgl. Tabelle 3-2). Unter dem Begriff Modellierung ist allgemein die Systembeschreibung und nicht ausschließlich die Modellierung von Simulationsmodellen zu verstehen.

3.4 Anforderungen

Modellierung	Simulation im Prozess	Informationsmanagement
1. Einheitliche Systembeschreibung	2. Parallelisierung von Aktivitäten 3. Frühzeitige Überprüfung von Systemwechselwirkungen 4. Durchgängiger Simulationseinsatz	5. Intuitive und schnelle Informationsbeschaffung 6. Intensivierung des Informationsaustausches
7. Gewährleistung der Umsetzbarkeit		

Tabelle 3-2 Zusammenfassung der Hauptanforderungen

Diese drei Bereiche sind nicht immer scharf voneinander zu trennen und überlappen sich zum Teil. In der anschließenden Konzeptentwicklung werden die Bereiche als Bausteine des Gesamtkonzepts betrachtet, die integriert schließlich das vollständige Prozessmodell ergeben. Im Folgenden werden die Anforderungen aus Tabelle 3-2 weiter detailliert.

Anforderung 1. Einheitliche Systembeschreibung

An der Entwicklung sind Experten verschiedener Fachabteilungen beteiligt, die unterschiedliche Werkzeuge, Methodiken und Begriffswelten nutzen (vgl. Defizit 3). Für den eindeutigen, unmissverständlichen Informationsaustausch wird eine standardisierte Systembeschreibung benötigt, die es ermöglicht, sämtliche an der Entwicklung beteiligten Fachdisziplinen allgemein darzustellen.

Anforderung 2. Parallelisierung von Aktivitäten

Zum gegenwärtigen Zeitpunkt ist die Zusammenarbeit zwischen verschiedenen Abteilungen gering: Das hat zur Folge, dass der Entwicklungsprozess sequenziell durchlaufen wird. Durch das Warten auf bestimmte Ergebnisse aus vorangegangenen Entwicklungsschritten kommt es zu Verzögerungen (vgl. Defizit 3). Aus diesem Grund ist eine Parallelisierung von Aktivitäten erforderlich. So könnte z. B. die Testabteilung die Testvorrichtung unter Zuhilfenahme der Modelle aus der Detaillierungsphase frühzeitig aufbauen, ohne auf die Modelle und Komponenten der Zulieferer warten zu müssen. Für die Parallelisierung von Aktivitäten müssen verschiedene Anforderungen erfüllt werden. Zum einen müssen in der Detaillierungsphase Modelle erstellt werden, die sich für den Einsatz in nachfolgenden Phasen eignen, und zum anderen muss ein Modellaustausch gewährleistet sein. Die Anforderungen an einen durchgängigen Einsatz von Simulationstechniken im Systementwicklungsprozess bedingen damit weitere Anforderungen an das Informationsmanagement.

Anforderung 3. Frühzeitige Überprüfung von Systemwechselwirkungen

Eine der wichtigsten Anforderungen an ein zukünftiges Konzept stellt die frühzeitige Überprüfung der Systeminteraktionen dar. Zur Identifikation von Systeminkompatibilitäten ist es erforderlich, systemübergreifende Wechselwirkungen bereits in der Konzept- und anschließenden Detaillierungsphase zu berücksichtigen, anstatt diese wie bisher erst während der Integrationstests zu überprüfen. Wechselwirkungen zwischen den Systemen können zum einen durch eine intensive Abstimmung zwischen den Entwicklungsabteilungen überprüft werden. Die Grundlage hierfür bildet die Einbindung und Umsetzung des entsprechenden Informationsmanagements (vgl. Anforderungen 5 und 6) in dem Prozess. Zum anderen ist eine virtuelle Überprüfung des Zusammenwirkens erforderlich und damit die Vorverlegung der Systeminteraktionen aus den Integrationstests in die Konzept- bzw. Detaillierungsphase [31], [91]. Dies hat zur Folge, dass Fehler bereits frühzeitig erkannt werden und somit zeit- und kostenintensive Änderungsschleifen reduziert werden können. Ein weiterer Vorteil besteht darin, dass durch die kostengünstige Softwareanalyse ein Teil der späteren Hardwaretests entfällt. So sollen bereits 2011 virtuelle Prototypen in Verbindung mit Hardware in the Loop 90 % aller physikalischen Prototypen ersetzen [86]. Für die Realisierung dieser Anforderung sind bestimmte Voraussetzungen zu erfüllen. So muss der Modellaustausch standardisiert werden, um Modelle in ein Simulationsprogramm zu integrieren. Des Weiteren müssen Simulationskopplungen (vgl. Kapitel 2.3.2) realisiert werden, die den Einsatz der geeignetsten Simulationsprogramme gewährleisten. Zusammenfassend lässt sich sagen, dass die frühzeitige Überprüfung der Systemwechselwirkungen zu einer Erhöhung der Entwicklungsqualität in der gesamten Entwicklung führt.

Anforderung 4. Durchgängiger Simulationseinsatz

Ein weiteres Defizit des derzeitigen Entwicklungsprozesses ist die Festlegung auf ein Konzept ohne vorherige Überprüfung von Alternativen in der frühen Entwicklungsphase. Für die Auswahl des am besten geeigneten Konzeptes ist die Unterstützung der Konzeptphase durch den Einsatz von Simulationstechniken erforderlich. Die Simulation ermöglicht den direkten Vergleich vieler Lösungsmöglichkeiten und somit auch die Weiterentwicklung des den Anforderungen entsprechenden besten Konzeptes. Ein weiterer Vorteil des frühen Einsatzes von Simulation besteht darin, dass das Reparaturverhalten reduziert wird, da die Ursache für Folgefehler, die auf einem fehlerhaften Konzept basieren, durch die optimale Konzeptauswahl behoben wurde. Der durchgängige Einsatz von Simulation stellt weitere Anforderungen, wie z. B. einen geeigneten Modellaustausch, eine standardisierte Schnittstelle usw., an ein zukünftiges Prozesskonzept.

Anforderung 5. Intuitive Informationsbeschaffung

Die Grundlage zur Realisierung der vorangegangenen Anforderungen bildet ein Informationsmanagementsystem, in dem die an der Entwicklung Beteiligten alle benötigten Modelle, Daten und sonstigen Informationen finden. Für eine Vereinfachung der Informationsbeschaffung, die Optimierung der Kommunikation und erhöhte Wissensteilung im Unternehmen ist diese Vernetzung der Informationsbereitstellung von großer Bedeutung. Zudem muss die Informationsbeschaffung für den Benutzer intuitiv und zeitnah erfolgen. Aus diesem Grund ist eine einheitliche Benutzerschnittstelle erforderlich, die dem Nutzer übersichtlich die für ihn relevanten Informationen bereitstellt, wobei individuelle Zugriffsrechte die Informationsüberflutung verhindern. Zur einfacheren Handhabung und zum besseren Verständnis ist eine Standardisierung der bereitgestellten Informationen erforderlich.

Anforderung 6. Intensivierung des Informationsaustausches

Für einen erhöhten Informationsaustausch ist die Erfüllung der fünften Anforderung von großer Bedeutung. Eine Parallelisierung von Aktivitäten sowie eine systemübergreifende Analyse von Wechselwirkungen sind ohne ein geeignetes Informationsmanagementsystem in einem internationalen Unternehmen, in dem verschiedene Fachabteilungen an der Systementwicklung beteiligt sind, nicht zu realisieren. Das Informationsmanagementsystem muss die Ablage von Informationen in Form von Dokumenten, Modellen, Simulationsergebnissen und Daten jeglicher Art gewährleisten. Damit verbunden ist die Bereitstellung von ausreichender Speicherkapazität. Für die Wiederverwendung ist es entscheidend, dass die Informationen festgelegten Standards entsprechen. Zusätzlich müssen die Informationen vor dem Zugriff von Unbefugten geschützt werden.

Für die Akzeptanz der Nutzer ist es wichtig, dass die standardisierte Ablage ihrer Informationen nicht einen erheblichen Mehraufwand bedeutet, da sonst die Informationen lokal gespeichert werden und somit nicht für die Wissensteilung im Unternehmen zur Verfügung stehen. Eine Voraussetzung für den Modell- bzw. Datenaustausch ist die vorgeschriebene Ablage im Informationsmanagementsystem zu festgelegten Zeitpunkten.

Die Informationen im Prozess müssen zeitnah, das heißt sobald verfügbar, kommuniziert werden, sodass die Entwicklung in allen Abteilungen auf Basis der aktuellsten Informationen durchgeführt wird. Aus diesem Grund ist ein entsprechendes Versionsmanagement durchzuführen, das dem Nutzer den Status der Informationen bekannt gibt. Zusätzlich sind die Qualität der Information (z. B. für Modelle: validiert, Modellabstraktionsgrad usw.) und die Quelle anzugeben. Diese Anforderungen führen zur Reduzierung von Missverständnissen und von damit verbundenem Mehraufwand. Für eine intensivere Nutzung von Simulationsergebnissen ist eine Dokumentation der Simulationsstudie erforderlich. Zudem sollten Möglichkeiten genutzt werden, einen direkten Datenaustausch zwischen Simulationsprogrammen durchzuführen. Zur Reduzierung des Modellierungsaufwands ist eine intensive Wiederverwendung von Modellen erforderlich. Voraussetzungen für den Modellaustausch sind dabei Schnittstellenstandards.

Bisher ist die Lieferung von Komponentenmodellen der Zulieferer nicht verbindlich geregelt, zukünftig muss dies der Fall sein, um eine konsistente Modellverwendung sicherzustellen.

Anforderung 7. Gewährleistung der Umsetzbarkeit

Um die Realisierung des zu konzipierenden Konzepts zum Einsatz von Simulation gewährleisten zu können, muss der Nutzen den Aufwand, den die Einführung eines solchen

3.4 Anforderungen

Konzepts nach sich zieht, rechtfertigen. Das Gleiche gilt für den kontinuierlichen Aufwand für die prozessbegleitenden Tätigkeiten. Mit Einstein gesprochen heißt das: „Mach alles so einfach wie möglich, aber nicht einfacher“ [24]. Im Bezug auf die Modellierung bedeutet dies, die Modelle so genau wie möglich und so komplex wie nötig zu entwickeln. Hierfür sind den Entwicklungsphasen angemessene Detaillierungsgrade festzulegen, d. h., die Modellkomplexität steigt mit vorschreitender Entwicklung an. Für den Einsatz von Simulationstechniken und systemübergreifenden Simulationen bedeutet es, dass Simulation nur dann eingesetzt wird, wenn sie von Nutzen ist. Nicht zu vernachlässigen sind die Investitionskosten für ein zukünftiges Konzept. Aus diesem Grund sollen vorhandene Ressourcen (z. B. Datenverwaltungssysteme, Softwareprogramme, Hardwarekomponenten usw.) adaptiert werden. Eine weitere Möglichkeit zur Reduzierung der Konzeptkosten bietet der Einsatz frei erhältlicher Software (Abk. GPL; engl.: General Public License) [56].

In Tabelle 3-3 sind die Anforderungen an ein zukünftiges Konzept dargestellt. Neben den bisher beschriebenen Anforderungen sind weitere selbsterklärende Anforderungen angeführt. Die Anforderungen sind in die vier Bereiche Modellierung, Simulationseinsatz im Prozess, Informationsmanagement und Sonstiges eingeteilt.

Analyse der Istsituation

Bereich	Anforderung
Modellierung	1. Einheitliche Systembeschreibung <ul style="list-style-type: none"> - Standardisierte Systembeschreibung - Modellierung heterogener physikalischer Systeme - Optimierung des Gesamtsystemverständnisses - Reduzierung von Missverständnissen
Simulation im Prozess	2. Parallelisierung von Aktivitäten <ul style="list-style-type: none"> - Modellwiederverwendung gewährleisten - Frühzeitige Erstellung der Testvorrichtung - Einheitliches globales Umgebungsmodell 3. Frühzeitige Überprüfung von Systemwechselwirkungen <ul style="list-style-type: none"> - Konsistente Modellierung und Simulation - Systemübergreifende Simulation gewährleisten - Schnittstellenstandards - Standardisierter Modellaustausch - Virtuelle Überprüfung der Systemwechselwirkungen - Systemübergreifende Entwicklungskoordination - Einfache Modellintegration - Reduzierung der Hardwaretests und Prototypen - Einsatz des am besten geeigneten Simulationsprogramms 4. Durchgängiger Simulationseinsatz <ul style="list-style-type: none"> - Unabhängige Validierung der Modelle - Simulationsgestützte Konzeptauswahl - Erstellung von Konzeptalternativen - Lieferung von Komponentenmodellen durch Zulieferer - Kontinuierlicher Simulationseinsatz - Modularisierung der Simulationsmodelle - Reduktion des Reparaturverhaltens - Hierarchisches Modellierungskonzept
Informationsmanagement	5. Intuitive Informationsbeschaffung <ul style="list-style-type: none"> - Einheitliche Benutzerschnittstelle - Einheitlicher zentraler Informationszugriff - Einfache Informationsbeschaffung - Verhinderung der Datenüberflutung - Automatische Benachrichtigung über relevante Informationen - Übersichtliche Verwaltungsstruktur 6. Intensivierung des Informationsaustausches <ul style="list-style-type: none"> - Standardisierte Dokumentation - Verbesserung der Kommunikation - Standardisierte Ablage von Simulationsergebnissen - Kontinuierliche Kommunikation von Informationen - Zeitnahe Informationsbereitstellung (Aktualität) - Statusangabe der Informationen (Qualität, Quelle) - Direkter Datenaustausch zwischen Simulationsprogrammen - Schnittstellenstandards - Unternehmensweite Bereitstellung der benötigten Informationen - Ausreichende Speicherkapazität - Kontrolle der Informationen - Individuelle Rechtevergabe - Konfigurationsmanagement und Versionsmanagement - Einfache Informationsablage (Verhindern lokaler Speicherung) - Versionsmanagement - Speicherung sowie Bereitstellung aller benötigten Informationen - Informationen vor dem Zugriff durch Unbefugte schützen - Informationsablage zu bestimmten Zeitpunkten regeln - Bereitstellung von Simulationsmodellen
Sonstiges	7. Gewährleistung der Umsetzbarkeit <ul style="list-style-type: none"> - Niedrige Investitionskosten - Weiterverwendung bestehender Ressourcen - Modellkomplexität den Entwicklungsphasen anpassen - Einsatz von GPL-Software - Adäquater Simulationseinsatz - Geringer Betreuungsaufwand des DBMS - Geringer Schulungsaufwand für die Mitarbeiter

Tabelle 3-3 Konzeptanforderungen

3.4 Anforderungen

Einige Anforderungen für den erfolgreichen Einsatz eines zukünftigen Konzepts zur durchgängigen Simulationsunterstützung im Flugzeugsystementwicklungsprozess können nicht mithilfe von Vorgehensweisen, Hilfsmitteln, Methodiken, Richtlinien usw. erfüllt werden. Das neue Konzept muss zusätzlich von den an der Entwicklung beteiligten Personen gelebt und verstanden werden, sie müssen z. B. bereit zur Wissensteilung im Unternehmen sein. Eine hohe Akzeptanz des neuen Konzeptes basiert auf den Anforderungen zur Gewährleistung der Umsetzbarkeit. Für den einzelnen Mitarbeiter ist hierbei die Weiterverwendung von bestehenden Ressourcen sowie ein geringer Betreuungs- und Schulungsaufwand von besonderer Bedeutung. Niedrige Investitionskosten bei der Einführung des Konzeptes spielen für das Unternehmensmanagement eine übergeordnete Rolle. Die Erfüllung der sonstigen Anforderungen (vgl. Anforderung 7) führt damit zu einer Erhöhung der Akzeptanz des neuen Konzeptes und maximiert somit dessen Erfolgsaussichten.

Ausgehend von den in diesem Kapitel hergeleiteten Anforderungen an ein zukünftiges Prozesskonzept werden im Folgendem Lösungen für die Erfüllung dieser Anforderungen beschrieben und verglichen. Die am besten geeigneten Lösungen werden anschließend zu einem Gesamtkonzept kombiniert.

4 Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

Die Besonderheiten in der Entwicklung von interdisziplinären und hochintegrierten Flugzeugsystemen stellen erhebliche Anforderungen an den Entwicklungsprozess. Bedingt durch den harten Konkurrenzkampf der Luftfahrtindustrie, werden diese zusätzlich verstärkt. Für die Aufrechterhaltung der Wettbewerbsfähigkeit sind die enge Verknüpfung von Mechanik, Elektro- und Regelungstechnik sowie eine methodische Vorgehensweise und die konsequente Nutzung der Informationstechnik von besonderer Bedeutung [183].

Die Computersimulation hat sich als ein hocheffizientes Kommunikationsmittel erwiesen, da sie die technischen Sachverhalte inklusive der verschiedenen Abhängigkeiten von Einzelsystemen für alle Beteiligten allgemein verständlich und frei von subjektiven Einflüssen visualisiert [9]. Zusätzlich führt der Einsatz von Simulationstechniken zur Leistungssteigerung in allen am Produktentstehungsprozess beteiligten Bereichen und unterstützt die Qualitätsoptimierung sowie die Minimierung von Zeit und Kosten (vgl. Kapitel 2.3). Die Frage ist nicht mehr, ob Simulationstechniken in der Produktentwicklung eingesetzt werden sollen, sondern wann und wie dieses zu geschehen hat.

Von einem durchgängigen Simulationseinsatz mit den beschriebenen Vorteilen ist man in der Flugzeugindustrie weit entfernt. Simulationstechniken kommen erst in der späten Konzeptphase und der Detaillierungsphase zum Einsatz und werden zur Analyse lokaler Teilsysteme verwendet, die positiven Auswirkungen auf den Entwicklungsprozess sind marginal. Es fehlt an einem Konzept zur durchgängigen Simulationsunterstützung in sämtlichen Entwicklungsphasen (vgl. Kapitel 2.3.4). Aus diesem Grund ist die Erstellung eines generischen Konzepts für den durchgängigen Simulationseinsatz erforderlich.

Ein geeignetes Konzept zeichnet sich durch die folgenden drei Eigenschaften aus (vgl. Tabelle 3-3): Die für die Simulation benötigten Informationen werden durch den Einsatz eines Informationsmanagements bereitgestellt. Der Einsatz der Simulationstechnik muss in allen Phasen geregelt sein, und aus Gründen der Schnittstellensynchronisation sowie zur Erhöhung der Transparenz wird eine geeignete Modellierung benötigt (siehe Abbildung 4-1).

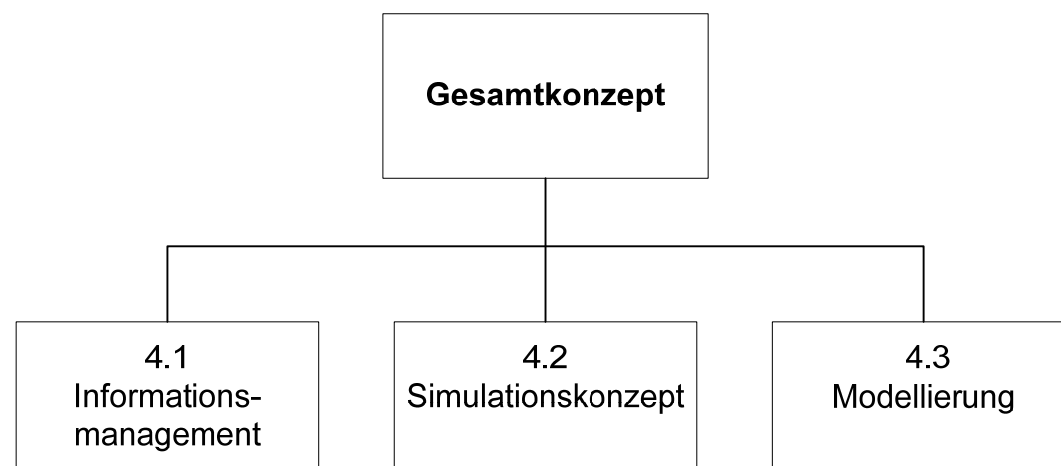


Abbildung 4-1 Die drei Pfeiler einer durchgängigen Simulationsunterstützung

Für die Simulation multidisziplinärer Systeme steht heute eine Vielzahl von Simulationswerkzeugen für sämtliche technischen Bereiche zur Verfügung. Zudem sind die Arbeitsplatzrechner so leistungsstark, dass selbst komplexe technische Systeme ohne den Einsatz von Großrechnern in angemessener Zeit simuliert werden können. Für die Bereitstellung von Informationen stehen anwendungsspezifische Lösungen zur Verfügung, und auch im Bereich der Modellierung existieren verschiedenste Standards. Damit sind sämtliche für die Umsetzung einer durchgängigen Simulationsunterstützung benötigten Technologien verfügbar. Für einen kontinuierlichen Einsatz der Simulationstechnik im Entwicklungsprozess ist ein Konzept erforderlich, welches regelt, wann und in welcher Form die Simulations- bzw. Informationstechnologie eingesetzt werden soll und welches Vorgehen bei der Modellierung verwendet werden soll. Dabei ist auf eine abstrakte bzw. generische Konzeptgestaltung zu achten, um so ein allgemeingültiges Vorgehen bereitzustellen. Im Folgenden wird ein

4.1 Simulationskonzept

Konzept erarbeitet, das diese Voraussetzungen erfüllt. Basierend auf den Anforderungen aus Kapitel 3, werden verschiedene Konzeptalternativen erstellt, verglichen und bewertet, um so das am besten geeignete Konzept zu ermitteln. Die Einzellösungen für die Modellierung, das Simulationskonzept und die Informationsbereitstellung werden anschließend zu einem Gesamtkonzept für die durchgängige Simulationsunterstützung im Flugzeugsystementwicklungsprozess (Abk. FSEP) kombiniert. Der in dieser Arbeit betrachtete Entwicklungsprozess beginnt mit der Definition der Hauptsystemanforderungen vor der Konzeptphase (M4–M5) und endet mit dem Flugtest (vgl. Abbildung 3-1).

4.1 Simulationskonzept

Der kontinuierliche Einsatz der Simulationstechnik in allen Entwicklungsphasen, von der Aufnahme der Anforderungen bis zum fertigen Produkt, erfordert ein konsistentes, systemübergreifendes Simulationskonzept. Des Weiteren muss die Einhaltung des kompletten Problesezyklus unterstützt werden (vgl. Kapitel 2.2.1), um z. B. Lösungsalternativen in den frühen Entwicklungsphasen zu vergleichen [186]. Für die Entwicklung eines durchgängigen Simulationskonzeptes sind aus diesem Grund einige Voraussetzungen, wie z. B. verschiedene Modellabstraktionsgrade und die Simulationskopplung, erforderlich (vgl. Kapitel 2.3.1 und 2.3.2). In diesem Kapitel werden zunächst verschiedene Simulationskonzeptalternativen auf die Erfüllung der Anforderungen hin überprüft.

4.1.1 Simulationskonzeptalternativen

Die Analyse verschiedener Simulationskonzeptalternativen ist aus dem gleichen Grund von Bedeutung wie der Einsatz von Simulationstechniken in der frühen Entwicklungsphase. Durch den direkten Vergleich von Konzeptalternativen kann gewährleistet werden, dass nicht die nächstbeste, sondern die beste Lösung gefunden wird [174]. Die Auswahl des Simulationskonzeptes erfolgt basierend auf den Anforderungen, mithilfe eines Kriterienkatalogs. Im Folgenden werden drei Grobkonzepte für die durchgängige Simulationsunterstützung vorgestellt. Der reibungslose Informationsaustausch wird dabei zunächst vorausgesetzt. Ein geeignetes Konzept für die Informationsbereitstellung wird in Kapitel 4.2 vorgestellt.

Konzeptalternative 1: alleinstehende Simulation

Im Entwicklungsprozess werden alleinstehende Simulationen durchgängig in allen Phasen eingesetzt. Architekturmodelle (vgl. Abbildung 2-7) mit geringer Komplexität werden in der frühen Konzeptphase zur Simulation verschiedener Systemarchitekturen eingesetzt. In der Detaillierungsphase werden die Teilsysteme wie bisher mit situationsbedingt ausgewählten Simulationsprogrammen simuliert. Eine systemübergreifende Gesamtsimulation der Teilsysteme ist nicht vorgesehen. Allerdings werden die Ergebnisdaten der Einzelsimulationen mithilfe des Informationsmanagements für alle an der Entwicklung Beteiligten zur Verfügung gestellt, sodass diese als Parameter für andere Teilsystems Simulationen genutzt werden können (vgl. Kapitel 2.3.2). Zusätzlich sind die Zulieferer dazu verpflichtet, parallel zur Hardware Simulationsmodelle der Komponenten zu liefern, die direkt in die Testvorrichtung eingebunden werden können. In der Testabteilung werden wie bisher die System-, Funktions- und Integrationstests durchgeführt. Der Unterschied zum bestehenden Prozess besteht darin, dass Simulationen durchgängig im Prozess, d. h. auch in den frühen Phasen der Konzeptfindung, eingesetzt werden. In Abbildung 4-2 ist das Konzept der alleinstehenden Simulationen dargestellt. Die Teilsimulationen können Simulationsdaten nicht direkt, sondern immer nur über das Informationsmanagement austauschen.

Ein Vorteil dieses Konzepts besteht darin, dass immer das am besten geeignete Simulationsprogramm für den jeweiligen Anwendungsfall eingesetzt werden kann. Zudem entfällt der Aufwand für den Aufbau einer Gesamtsimulation der Teilsysteme. Allerdings erfordert die Wiederverwendung von Modellen in der Testabteilung die Bereitstellung von Schnittstellen für z. B. HIL-Simulationen.

Der größte Nachteil dieses Konzepts besteht darin, dass der Einsatz von Simulation nur beschränkt auf ein Teilsystem stattfindet und damit die beteiligten Entwicklungsabteilungen die Teilsysteme unabhängig voneinander entwickeln. Dies führt dazu, dass das ganzheitliche Denken der beteiligten Entwickler eingeschränkt wird.

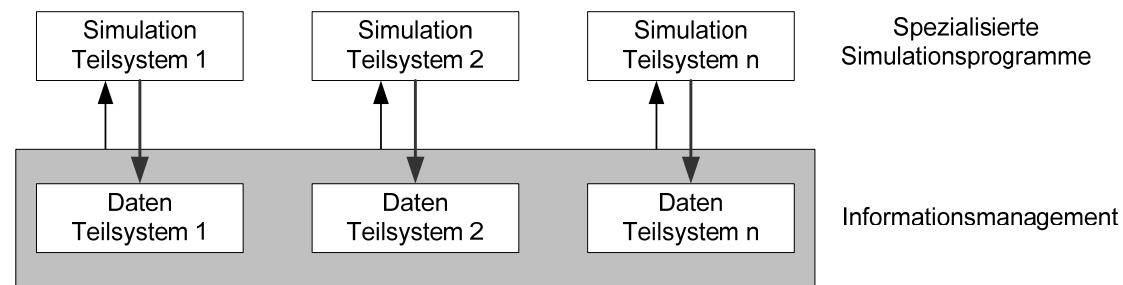


Abbildung 4-2 Alleinstehende Simulation der Teilsysteme

Der Austausch von Simulationsdaten wird zwar vom Informationsmanagement unterstützt, allerdings erlaubt die sequenzielle Simulation der Teilsysteme (vgl. Abbildung 2-9) nur den Austausch von Parametern. Ein direkter Variablentransfer zwischen den Teilsimulationen während der Laufzeit ist nicht möglich. Aus diesem Grund bleiben dynamische Wechselwirkungen zwischen den Teilsystemen unberücksichtigt, was zu Inkompatibilitäten führen kann.

Konzeptalternative 2: durchgängig systemübergreifende Simulation

Die durchgängige Simulationsunterstützung wird durch den Einsatz einer multidisziplinären, systemübergreifenden Simulationsumgebung erreicht, für deren Realisierung drei domänenübergreifende Simulationsvarianten verwendet werden können (vgl. Abbildung 2-8). Es sei darauf hingewiesen, dass die lose Kopplung für diese Konzeptalternative nicht in Betracht gezogen wird.

Der Aufwand für die Implementierung der systemübergreifenden Simulation hängt im Wesentlichen von der verwendeten Simulationsvariante ab. Weisen die an der Simulation beteiligten Simulationsprogramme z. B. Schnittstellen für die Kosimulation auf, ist der Aufwand für die Erstellung einer systemübergreifenden Simulation im Vergleich zum Aufbau einer verteilten Simulation gering. Allerdings ist die Wahrscheinlichkeit, dass aus dem großen Angebot an Simulationsprogrammen gerade die verwendeten Programme Kosimulationsschnittstellen besitzen, gering.

Diese Konzeptalternative zeichnet sich durch eine kontinuierliche Simulation aller Systeminteraktionen aus. So wird in der Konzeptphase die systemübergreifende Simulation eingesetzt, um die Auswahl der Systemarchitektur auf einer groben Detaillierungsebene zu unterstützen. Wie in Konzept 1 beschrieben steigen mit voranschreitendem Entwicklungsprozess die gewonnenen Erkenntnisse über das Gesamtsystemverhalten und die Komplexität der Modelle an. Proportional dazu wachsen aber auch der Aufwand zur Erstellung der systemübergreifenden Simulation sowie die Simulationszeit an. Die Systementwicklung wird bis hin zur Spezifikationsebene durchgängig multidisziplinär simuliert. Nach Abschluss der Komponentenentwicklung liefert der Zulieferer neben der Hardware ebenfalls Simulationsmodelle, die bei Bedarf in die multidisziplinäre Simulationsumgebung bzw. in die Testumgebung eingebunden werden können. Zusätzlich kann die Testabteilung bereits während der Komponentenentwicklung auf die Modelle der Systemabteilung zugreifen, um frühzeitig mit dem Aufbau der Testvorrichtung beginnen zu können. Die multidisziplinäre Simulation wird durch ein systemübergreifendes Informationsmanagement ergänzt. Die Überwachung und Kontrolle der systemübergreifenden Simulationen wird von einer Koordinationsfunktion, die im Unternehmen implementiert sein muss, durchgeführt. Abbildung 4-3 zeigt den Aufbau einer systemübergreifenden Simulationsumgebung. In der Simulationsumgebung können entweder Modelle integriert werden und dann zusammen simuliert werden (vgl. Varianten 1 und 2 in Abbildung 2-8) oder es kann eine Simulationskopplung vorgenommen werden (vgl. Variante 3).

Im Gegensatz zum ersten Konzept werden durch den Einsatz einer systemübergreifenden Simulation die Wechselwirkungen der Teilsysteme frühzeitig und nicht erst in den System- und Integrationstests berücksichtigt. Für die an der Entwicklung Beteiligten bedeutet dies eine Verbesserung des ganzheitlichen Systemverständnisses.

4.1 Simulationskonzept

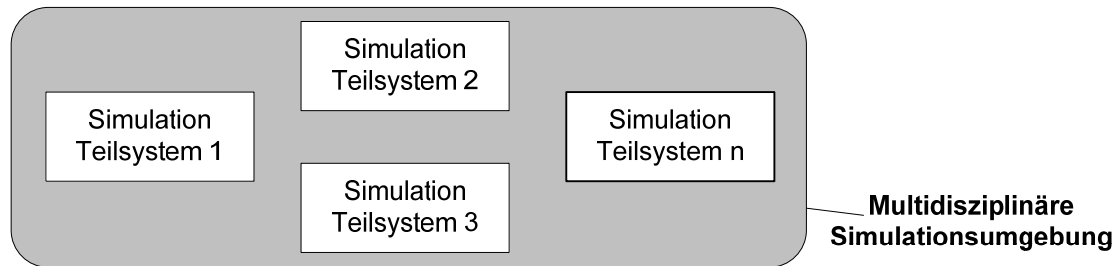


Abbildung 4-3 Durchgängig systemübergreifende Simulation

Neben diesen Vorteilen hat eine durchgängig systemübergreifende Simulation jedoch eine Reihe von Nachteilen. Zum einen ist die Abstimmung zwischen den an der Entwicklung beteiligten Mitarbeitern aufwendig, zum anderen erfordert die Erstellung einer multidisziplinären Simulationsumgebung einen hohen Ressourceneinsatz.

Konzeptalternative 3: Kombination aus alleinstehenden und systemübergreifenden Simulationen

Wie schon in den beiden vorangegangenen Konzepten wird die Simulation durchgängig in allen Entwicklungsphasen eingesetzt. Da es sich um eine Kombination der ersten beiden Konzeptalternativen handelt, erfolgt der Einsatz der Simulation analog zu diesen. Der Unterschied besteht darin, dass die systemübergreifende Simulation des Gesamtsystems nicht kontinuierlich vorgenommen wird, sondern zu bestimmten Zeitpunkten im Entwicklungsprozess (Meilensteinen). Zwischen den systemübergreifenden Simulationen werden wie bisher alleinstehende Simulationen der Teilsysteme durchgeführt. Für den Einsatz der systemübergreifenden Simulation eignen sich das Ende der Konzeptentwicklung sowie das Ende der Detaillierungsphase (vgl. Abbildung 3-1), da hier das Konzept bzw. die Spezifikation festgelegt werden. Simulationsmodelle aus den Systemabteilungen werden parallel zur Komponentenentwicklung an die Testabteilung weitergegeben, sodass die Testvorrichtung frühzeitig erstellt werden kann. Im Rahmen der Hardware in the Loop-Tests können bei Bedarf Simulationsmodelle aus den vorangegangenen Simulationen eingesetzt werden. Die Basis für den Informationsaustausch bildet wie in der vorangegangenen Konzeptalternativen das Informationsmanagement. Die Koordinierung der Simulationen wird wiederum von einer Koordinationsfunktion, die im Unternehmen bereitgestellt wird, übernommen.

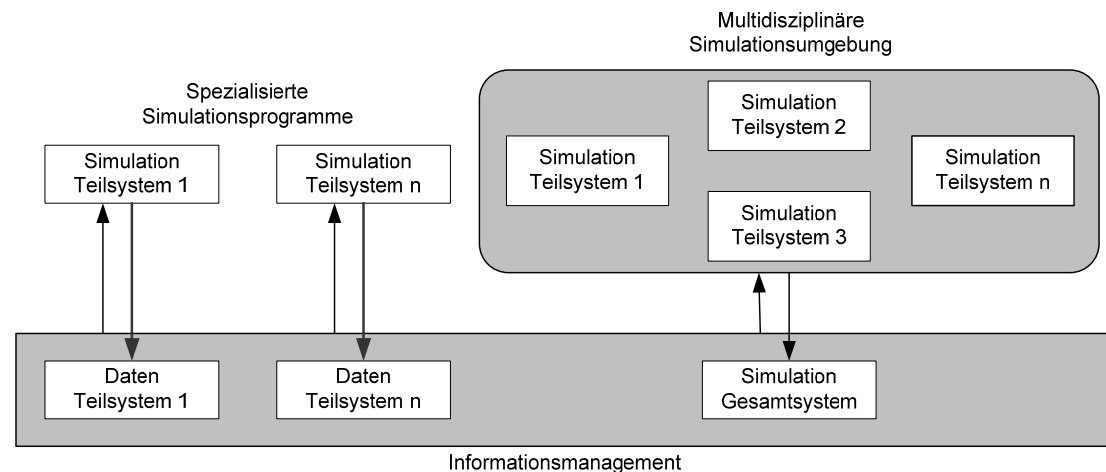


Abbildung 4-4 Kombination aus alleinstehenden und systemübergreifenden Simulationen

Die dritte Konzeptalternative vereint die Vorteile der beiden vorangegangenen Konzepte. Zum einen werden die Systeminteraktionen in der Simulation berücksichtigt, und zum anderen werden alleinstehende und damit ressourcenschonende Simulationen von Teilsystemen unterstützt. Zudem sind alleinstehende Simulationen flexibler und können einfacher an veränderte Bedingungen angepasst werden. Zusätzlich können Modellierungsfehler leichter identifiziert werden als in einer Gesamtsimulation. Aus diesem Grund kann sich bei einer eventuellen Fehlersuche in der Gesamtsimulation auf die Schnittstellen konzentriert werden. Ein weiterer Vorteil besteht darin, dass die Teilsysteme bereits vor der systemübergreifenden Simulation einen Optimierungsprozess durchlaufen haben, wodurch sich die Gesamtsystem-

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

optimierung vereinfacht. Die Komplexität in der Entwicklung, der Simulationsaufwand und somit der Ressourceneinsatz werden in Relation zum Nutzen der frühen Erkennung von Inkompatibilitäten und der rechtzeitigen Durchführung von Änderungen auf einem angemessenen Maß gehalten. Der Aufwand für ein Vorgehen nach diesem Konzept ist somit höher als beim ersten, aber nicht so hoch wie beim zweiten Konzept.

In der folgenden Tabelle 4-1 werden verschiedene Kriterien, die sich an den Anforderungen aus Kapitel 3 orientieren, der drei vorgestellten Konzeptalternativen bewertet. Für die Bewertung der Kriterien kommt ein dreistufiges Bewertungsverfahren zum Einsatz. Dabei steht ein „+“ für ein erfülltes, eine „0“ für eine teilweise erfülltes und ein „-“ für ein nicht erfülltes Kriterium.

Kriterien	Konzept 1 Alleinstehend	Konzept 2 System-über- greifend	Konzept 3 Kombination
Parallelisierung von Aktivitäten	+	+	+
Durchgängige Überprüfung der Systemwechselwirkungen	-	+	+
Analyse von Systemarchitekturen in der Konzeptphase	0	+	+
Ressourcenaufwand	+	-	0

Tabelle 4-1 Bewertung der Grobkonzepte für den Simulationseinsatz

Die Wiederverwendung von Modellen sowie der frühzeitige Aufbau der Testvorrichtungen werden von den drei Konzepten gleichermaßen unterstützt. Gegenüber den anderen Konzepten hat der Einsatz von alleinstehenden Simulationen den Vorteil, dass die Schnittstellen mit anderen Teilsystemen nicht berücksichtigt werden müssen. Hieraus ergibt sich der vergleichsweise geringe Aufwand zur Erstellung der Simulationen. Gleichzeitig resultiert aus diesem Vorteil aber auch der größte Nachteil des ersten Konzepts. Die Analyse des Gesamtsystemverhaltens ist unbefriedigend. Zudem findet eine erste Verifizierung der Teilsysteme unter Berücksichtigung der Verknüpfungen zu anderen Teilsystemen erst während der Integration statt, sodass bei Inkompatibilitäten späte Änderungen nötig werden, welche die Entwicklungszeit und -kosten negativ beeinflussen.

Die Konzepte zwei und drei weisen diesen Nachteil nicht auf. Systemwechselwirkungen werden frühzeitig (im linken Strang des V-Modells) berücksichtigt, sodass identifizierte Inkompatibilitäten mit geringem Zeit- und Kostenaufwand beseitigt werden können. Der Unterschied zum ersten Konzept besteht darin, dass systemübergreifende Simulationen bzw. eine Kombination aus den ersten beiden Konzepten eine Erhöhung des Ressourcenaufwands zur Folge haben. Das Konzept für den durchgängigen Einsatz multidisziplinärer Simulationen erfordert den höchsten Ressourceneinsatz. Durch die Gesamtsystems simulation wird zwar das Gesamtverständnis der an der Entwicklung Beteiligten gefördert, allerdings erschwert sich die Fehlersuche im Fall eines Fehlverhaltens. Die Kombination aus alleinstehenden und systemübergreifenden Simulationen weist die höchste Bewertung auf. In den systemübergreifenden Simulationen kommen bereits erprobte Modelle aus den vorangegangenen alleinstehenden Simulationen zum Einsatz. Bei einem Fehlverhalten der Simulation kann somit der Fokus der Fehlersuche auf die Schnittstellen der Simulation gerichtet werden. In diesem Konzept vereint sich das Teilsystemverständnis der beteiligten Entwicklungsabteilungen, welches auf den alleinstehenden Simulationen basiert, mit dem Gesamtsystemverständnis. In der systemübergreifenden Analyse ist es damit möglich, Teilsystemverhalten zu identifizieren und dieses hinsichtlich der Gesamtsystemoptimierung anzupassen. Hieraus ergibt sich, dass nicht die Summe der besten Teilsysteme das beste Gesamtsystem ergibt, vielmehr geht die Gesamtsystemoptimierung aus einer Kombination der im Bezug auf das Gesamtsystemverhalten optimalen Teilsysteme hervor.

Im Folgenden wird aus diesen Gründen die Detaillierung des dritten Konzeptes beschrieben. Um den Ressourceneinsatz für die multidisziplinäre Simulation auf einem angemessenen Niveau zu halten, sind allerdings Maßnahmen zu treffen, die den Prozess unterstützen und den Aufwand reduzieren.

4.1 Simulationskonzept

4.1.2 Detaillierung des Simulationskonzeptes

Durch den Einsatz des Simulationskonzeptes soll die kontinuierliche Abstimmung der Entwicklungsergebnisse über Systemgrenzen hinweg gewährleistet werden. Die Basis für die systemübergreifende Entwicklungsabstimmung und den damit verbundenen Datenaustausch bildet ein Informationsmanagement, welches in Kapitel 4.2 spezifiziert wird und in diesem Abschnitt zunächst als vorhanden vorausgesetzt wird. Die Simulation dient als Entscheidungshilfe, zur Visualisierung von Ergebnissen und zur Identifizierung von Inkompatibilitäten während der gesamten Systementwicklung. Zudem verbessert sich durch ihren Einsatz die Kommunikation zwischen den beteiligten Entwicklungsabteilungen [9], [55]. Ziel ist auch hier eine Erhöhung der Qualität bei gleichzeitiger Reduzierung der Entwicklungskosten und -zeit. Basierend auf der Kombination aus alleinstehenden und systemübergreifenden Simulationen erfolgt in diesem Kapitel eine Ausgestaltung des Grobkonzeptes.

4.1.2.1 Kontinuierlicher Simulationseinsatz

Eine der wichtigsten Anforderungen an das Simulationskonzept ist der durchgängige Einsatz von Simulation von der Konzeptentwicklung bis zur Inbetriebnahme des Flugzeugs (vgl. Abbildung 3-1). In Kapitel 2.3 wurde beschrieben, dass der Einsatz von Simulationstechniken in den frühen Prozessphasen von besonderer Bedeutung ist. Ein früher Einsatz von Simulation im Prozess ermöglicht es, durch die virtuelle Überprüfung der Entwicklung Fehler früh zu erkennen, sodass die Fehlerfolgekosten niedriger sind und die Qualität der Entwicklung insgesamt steigt. Des Weiteren wird wie bisher die Simulation als Hilfsmittel zur Detaillierung und Spezifizierung des Systems eingesetzt. Zusätzlich fördert die Erstellung und Lieferung von Komponentenmodellen der Zulieferer die durchgängige Modellverwendung sowie den konsistenten Informationsfluss. Eine weitere Reduzierung der Entwicklungszeit kann durch die Weitergabe von Modellen auf Spezifikationsebene (M6–M7, Verhaltensmodelle; vgl. Abbildung 2-7) von den Fachabteilungen an die Testabteilung erreicht werden.

1. Verstärkter Einsatz von Simulation in der frühen Konzeptphase

Für die Konzeptfindung zwischen den Meilensteinen M4 und M5 werden bislang aus Gründen der Ressourcenschonung wenig bis gar keine Simulationstechniken eingesetzt (vgl. Kapitel 3.1). Eine simulationsgestützte Analyse verschiedener Konzeptalternativen findet nicht statt. Dieses Vorgehen ist nicht angemessen, da nicht gewährleistet werden kann, dass ein den Anforderungen entsprechendes, optimales Konzept für die Weiterentwicklung ausgewählt wird. Der Prozess ist in den frühen Entwicklungsphasen starr und basiert auf empirischem Wissen, dies hat zur Folge, dass neue Technologienweiterentwicklungen nur sehr schleppend in der Flugzeugsystementwicklung eingeführt werden. Ziel dieses Simulationskonzeptes ist die virtuelle Analyse von Systemkonzeptalternativen, die im Gegensatz zu realen Tests eine schnelle Variation der Rahmenbedingungen sowie des gesamten Konzepts ermöglicht. Unkonventionelle Ideen, die ansonsten aufgrund fehlender Bewertungsmöglichkeit nicht weiter untersucht werden, können somit schnell auf Machbarkeit und Funktionsfähigkeit hin überprüft werden. Der frühzeitige Simulationseinsatz führt somit zu einer Erhöhung der Flexibilität in der Konzeptauswahl, sodass das beste – und nicht das erstbeste – Konzept ausgewählt wird.

Schwierigkeiten, die zu einer Behinderung des Simulationseinsatzes in den frühen Phasen der Entwicklung führen, basieren häufig auf geringem und zum Teil ungenauem Datenmaterial, wodurch der Modellaufbau sowie die Modellvalidierung erschwert oder sogar ganz verhindert werden. Ein weiterer Grund ist, dass Ressourcen wie Arbeitskräfte und Zeit in der frühen Konzeptphase beschränkt sind und den Simulationseinsatz limitieren, da besonders die Erstellung von neuen Simulationsmodellen sowie die Datenerhebung Ressourcen binden.

Die benötigten Daten für die Konzeptsimulationen basieren zum Teil auf den Systemanforderungen. Weitere Informationen, wie z. B. die Randbedingungen anderer Systeme, werden vom Informationsmanagementsystem zur Verfügung gestellt. Die Daten früherer Entwicklungen, zum Beispiel Erfahrungswerte und Modelle, sind ebenfalls im Informationsmanagementsystem hinterlegt und ermöglichen so den schnellen Aufbau von Architekturmodellen. Durch die Bereitstellung von Informationen und durch Wiederverwendung vorhandener Modelle wird auch bei einem begrenzten Ressourceneinsatz der Simulationseinsatz in den frühen Phasen der Entwicklung ermöglicht. Allerdings führt der intensive Einsatz von Simulationen in der frühen Entwicklungsphase zu einem Anstieg der Kosten im Ver-

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

gleich zum derzeitigen Entwicklungsprozess. Die Gesamtentwicklungskosten für den durchgängigen Einsatz der Simulationstechnik liegen allerdings unter den Kosten der derzeitigen Entwicklung. Diese Tatsache ist auf die frühzeitige Identifikation von Entwicklungsfehlern sowie die Auswahl der am besten geeigneten Konzeptalternative zurückzuführen. In Abbildung 4-5 ist der kumulative Entwicklungskostenverlauf als Funktion der Entwicklungszeit aufgetragen.

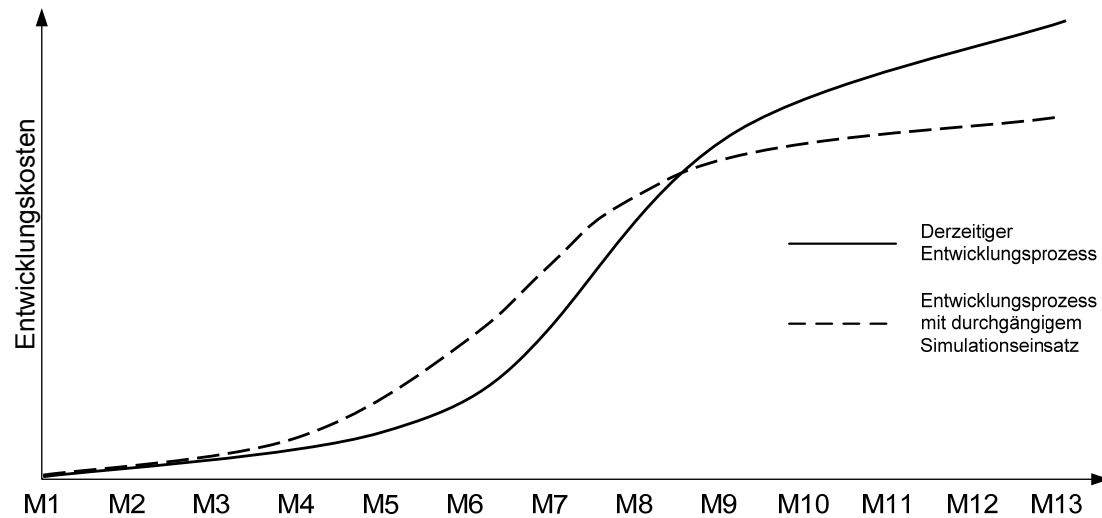


Abbildung 4-5 Entwicklungskosten für den Entwicklungsprozess mit und ohne durchgängigen Simulationseinsatz

Die Steigung der Kurve spiegelt die Höhe der Entwicklungskosten in den einzelnen Phasen wider. Eine starke Steigung ist dabei gleichbedeutend mit hohen Entwicklungskosten. Die höchsten Entwicklungskosten des derzeitigen Entwicklungsprozesses liegen im Bereich der Hardwareentwicklung (M7–M8), da in dieser Entwicklungsphase die Hardwarebeschaffungskosten anfallen. Spezifikationsfehler wirken sich auf diese Phase besonders kritisch aus und können zu zusätzlichen Hardwareinvestitionskosten führen. Aus diesem Grund ist die Steigung des Entwicklungskostenverlaufs mit einer durchgängigen Simulationsunterstützung in diesem Bereich wesentlich flacher als der derzeitige Verlauf, da Fehlentwicklungen früher erkannt werden.

2. Konsistente Weiterführung der Simulation bei den Zulieferern

Für die Realisierung eines durchgängigen Simulationskonzeptes müssen die Zulieferer ebenfalls in den Simulationsprozess integriert werden. Es muss vertraglich geregelt werden, dass die Zulieferer zusätzlich zur Hardware zeitgleich Softwaremodelle ihrer Systemkomponenten an den Flugzeughersteller liefern (M8), damit diese direkt in die Testvorrichtung eingebunden werden können und nicht von der Testabteilung eigenständig und mit Mehraufwand erstellt werden müssen. Werden in den anschließenden Tests Entwicklungsfehler entdeckt, hat dies oftmals zur Folge, dass Zulieferer den Entwicklungsprozess ebenfalls erneut durchlaufen müssen. In diesem Fall muss vertraglich gesichert sein, dass auch aktualisierte Simulationsmodelle erstellt und ausgeliefert werden. Gleichzeitig ist es wichtig, dass Änderungen an der Spezifikation, zum Beispiel durch eine frühere Verifizierung der Entwicklungsergebnisse, reduziert werden, um die Entwicklungssicherheit beim Zulieferer zu erhöhen. Die Nutzung der Simulationsmodelle der Zulieferer bedeutet eine Aufwandsreduzierung für die Testabteilung und gewährleistet Konsistenz. Für die reibungslose Modellwiederverwendung in der Testabteilung müssen Modellschnittstellen und Austauschformate festgelegt werden. Zusätzlich müssen dem Zulieferer die aktuellsten Entwicklungsinformationen zur Verfügung stehen. Ein beschränkter Zugriff auf das Informationsmanagementsystem (siehe Kapitel 4.2.2.3), gewährleistet die schnelle Entwicklungsanpassung beim Zulieferer. Zusätzlich stehen die aktuellsten Daten für die Entwicklung beim Zulieferer bereit, sodass z. B. erforderliche Randbedingungen für die Simulation oder Umgebungsmodelle aus vorangegangenen Entwicklungsphasen schnell gefunden bzw. wiederverwendet werden können. Die Vergabe der Zugangsrechte des Informationsmanagementsystems sowie die Synchronisation der Schnittstellen und Austauschformate werden von

4.1 Simulationskonzept

einer speziell dafür im Unternehmen implementierten Koordinationsfunktion kontrolliert, der Aufbau und die Aufgaben dieser Funktion werden im Kapitel 4.1.2.2 beschrieben.

3. Parallelisierung von Aktivitäten

Die Wiederverwendung von Modellen der Entwicklungsabteilungen in den Testabteilungen führt zu einer Reduktion von Aufwand und Entwicklungszeit. Mit Abschluss der Detaillierungsphase (M6–M7) legen die Fachabteilungen die Modelle, Dokumentationen und Daten aus der Detaillierungsphase im Informationsmanagementsystem ab, sodass die Testabteilungen auf die Daten der Fachabteilung zugreifen können. Spätestens mit Abschluss der Detaillierungsphase sind die Simulationsmodelle auf Detaillierungsebene (Verhaltensmodelle; vgl. Abbildung 2-7) im Informationsmanagementsystem verfügbar und können von der Testabteilung zur Erstellung der Testvorrichtung genutzt werden. Für HIL-Simulationen ist darauf zu achten, dass die eingesetzten Modelle echtzeitfähig sind. Weitere Modelle, die zum Beispiel aus den übergreifenden Simulationen stammen, können von der Testabteilung ebenfalls bei Bedarf genutzt werden. Somit erhält die Testabteilung parallel zur Komponentenentwicklung (M7–M8) frühzeitig die Möglichkeit, relevante Informationen zu nutzen und in den Testvorrichtungen umzusetzen.

4.1.2.2 Systemübergreifende Simulation

Bei komplexen Systemen ist es schon lange nicht mehr möglich, das Gesamtsystemverhalten aus der Summe des Teilsystemverhaltes abzuleiten. Es wird heute ein ganzheitliches Verständnis benötigt, das durch den Einsatz von Simulationstechniken erreicht werden kann.

Im bestehenden Prozess erfolgt die Verifizierung von Systemanforderungen erst im rechten Strang des V-Modells. „Es ist offensichtlich, dass dieser Prozess Entwicklungsfehler, die sich frühzeitig feststellen lassen, geradezu heraufbeschwört“ [88]. In Abbildung 4-6 sind die Fehler sowie deren Entdeckung als Funktionen der Zeit aufgetragen. Es wird deutlich, dass ein Großteil der Fehler aus frühen Entwicklungsphasen stammt und sich durch die Detektion in späten Entwicklungsphasen negativ auf die Entwicklungskosten auswirkt.

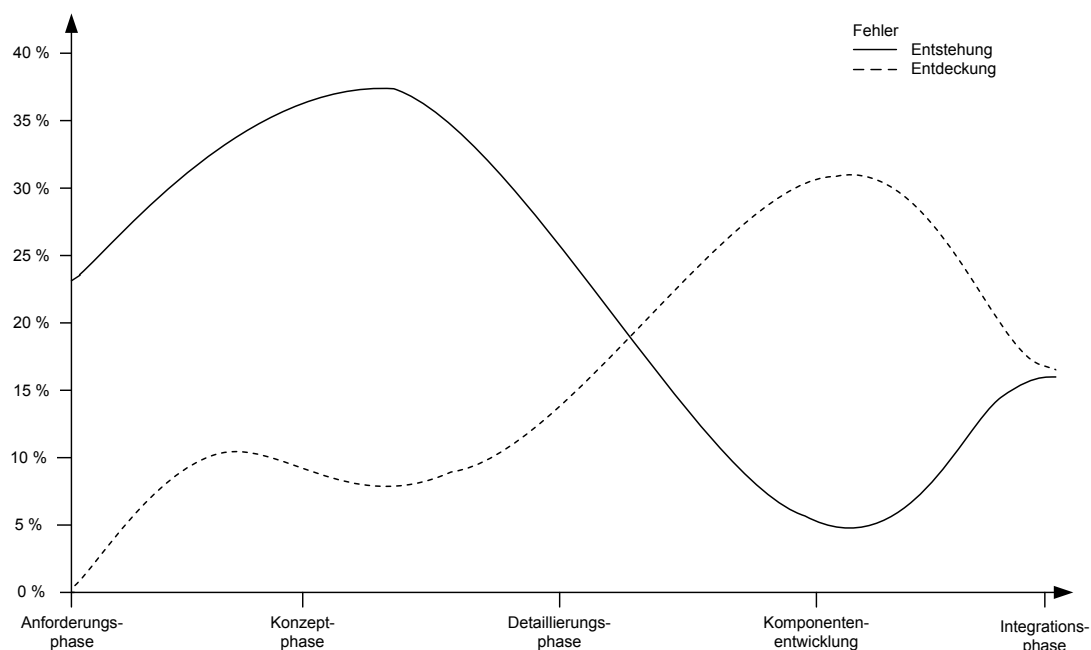


Abbildung 4-6 Fehler und Fehlerentdeckung im derzeitigen Entwicklungsprozess [88]

Die Hauptsystemanforderungen sind zum Teil nicht auf einzelne Entwicklungsbereiche oder Fachabteilungen begrenzt. Die Realisierung einer Anforderung, wie z. B. der Bereitstellung einer bestimmten Kabinentemperatur, weist eine Vielzahl von Abhängigkeiten auf. Sie ist abhängig von der Kabinengestaltung hinsichtlich Form und Materialien und von den Wärmeströmen durch die Kabine, die von den elektrischen Geräten und den Passagieren verursacht werden. Zudem ist die Bereitstellung einer gewissen Temperatur abhängig von den zugeführten Heizströmen und elektrischen Kapazitäten. Diese Reihe von Abhängigkeiten zeigt, dass zur Verifizierung einer einzelnen Anforderung systemübergreifende Beeinflussungen

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

berücksichtigt werden müssen. Die Abstimmung zwischen den Teilsystemen bildet die Basis für eine erfolgreiche Bereitstellung der geforderten Funktionen bzw. Anforderung. Voraussetzung hierfür ist ein intensiver Austausch von Informationen zwischen den beteiligten Systemabteilungen. Eine abteilungsübergreifende Entwicklung hat grundsätzlich den Vorteil, dass die miteinander verknüpften Systeme besser aufeinander abgestimmt und Inkompatibilitäten früher, nicht erst in den Integrationstests, erkannt werden (siehe Kapitel 2.2). Findet die systemübergreifende Entwicklung ausschließlich über einen verbesserten Informationsaustausch statt, besteht die Gefahr von Missverständnissen. Durch den Einsatz systemübergreifender Simulationen ist es möglich, den abteilungsübergreifenden Stand der Entwicklung ohne den Einsatz realer Tests virtuell zu überprüfen und zu verifizieren. Der Vorteil einer frühen systemübergreifenden Überprüfung der Konzepte liegt darin, dass Unstimmigkeiten zwischen den Systemkonzepten früh erkannt werden, sodass die Entwicklungsqualität gesteigert werden kann. Da Änderungen in diesen frühen Phasen der Entwicklung geringere Folgekosten als in späteren Phasen verursachen, sinken zudem die Entwicklungskosten (vgl. Abschnitt 2.2.2). Allerdings erfordert der Aufbau einer systemübergreifenden Simulation einen zusätzlichen Einsatz von Ressourcen. Aus diesem Grund sollte die systemübergreifende Simulation, wie bereits bei der Auswahl der Grobkonzepte beschrieben, nicht durchgängig, sondern nur zielgerichtet eingesetzt werden.

Für die systemübergreifende Simulation bieten sich zwei Zeitpunkte im Entwicklungsprozess an. Zum einen sollte die Konzeptfindung am Ende der Konzeptphase (M5) durch systemübergreifende Simulationen unterstützt werden, da hier das Konzept für die weitere Detaillierung festgelegt wird. Zum anderen sollte die Spezifikation (M6) durch den Einsatz der Simulation abgesichert werden, da auf Basis der Ergebnisse der Detaillierungsphase die Spezifikationen erstellt werden. Durch den Einsatz der systemübergreifenden Simulation wird die Entscheidungsqualität erheblich verbessert, sodass die Erfolgsaussichten der Entwicklung erhöht werden. Änderungen sowohl an der Systemarchitektur als auch an den Spezifikationen hätten große und aufwendige Iterationsschleifen im V-Modell zur Folge. Daher ist es sinnvoll, sowohl die Systemarchitektur als auch die Spezifizierung auf systemübergreifende Wechselwirkungen hin zu überprüfen, um Inkompatibilitäten frühzeitig beheben zu können.

In der Abbildung 4-7 ist das Vorgehen in der Entwicklung dargestellt, bei dem zum einen am Ende der Konzeptphase, vor Erstellen des Systemanforderungsdokuments (M5), und zum anderen am Ende der Definitionsphase, vor Erstellen der technischen Spezifikation (M6), systemübergreifende Simulationen durchgeführt werden. Nach Beendigung der systemübergreifenden Simulation müssen von den Entwicklungsabteilungen entweder Änderungen an der bisherigen Planung vorgenommen oder die nächste Entwicklungsphase (Detaillierung bzw. Komponentenentwicklung) freigegeben werden.

Zur Durchführung von systemübergreifenden Simulationen sind entsprechende Voraussetzungen zu schaffen. Erforderlich ist eine Koordination und Überwachung der gesamten Entwicklungsdaten der beteiligten Abteilungen. Aus diesem Grund ist eine Koordinationsfunktion im Unternehmen zu implementieren. Die Umsetzung dieser Funktion kann dabei z. B. durch ein Koordinationsteam oder eine IT-Lösung realisiert werden, wobei die Basis für diese Funktion wiederum von der Informationsbereitstellung gebildet wird. In der Koordinationsfunktion müssen weitere Unterfunktionen implementiert sein:

- Administrative Aufgaben, z. B. die Vergabe von Zugriffsrechten auf Daten
- Überprüfung der abgelegten Daten auf Vollständigkeit und Korrektheit
- Festlegung von Richtlinien für Modellschnittstellen, Austauschformate und die Datenverwaltung
- Speicherverwaltung
- Validierung und Verifikation der Modelle

Die wichtigste Voraussetzung für die systemübergreifende Simulation bildet die Modellschnittstelle. Hierfür existieren bereits verschiedene Industrierichtlinien in denen der Modellaustausch, die Schnittstellen sowie die Modellintegration spezifiziert sind [95]. Die Ziele dieser Richtlinien sind es, den Austausch und die Integration von Teilmodellen zu fördern sowie die Ausführung von Modellen in jeder geeigneten Simulationsumgebung zu ermöglichen. Bim derzeitigen Stand der Technik bietet sich die Verwendung von C-Code an, da in nahezu allen Simulationswerkzeugen eine C-Code-Schnittstelle implementiert ist. Die Festlegung der Richtlinien sowie deren Umsetzung sind Aufgaben der Koordinationsfunktion.

4.1 Simulationskonzept

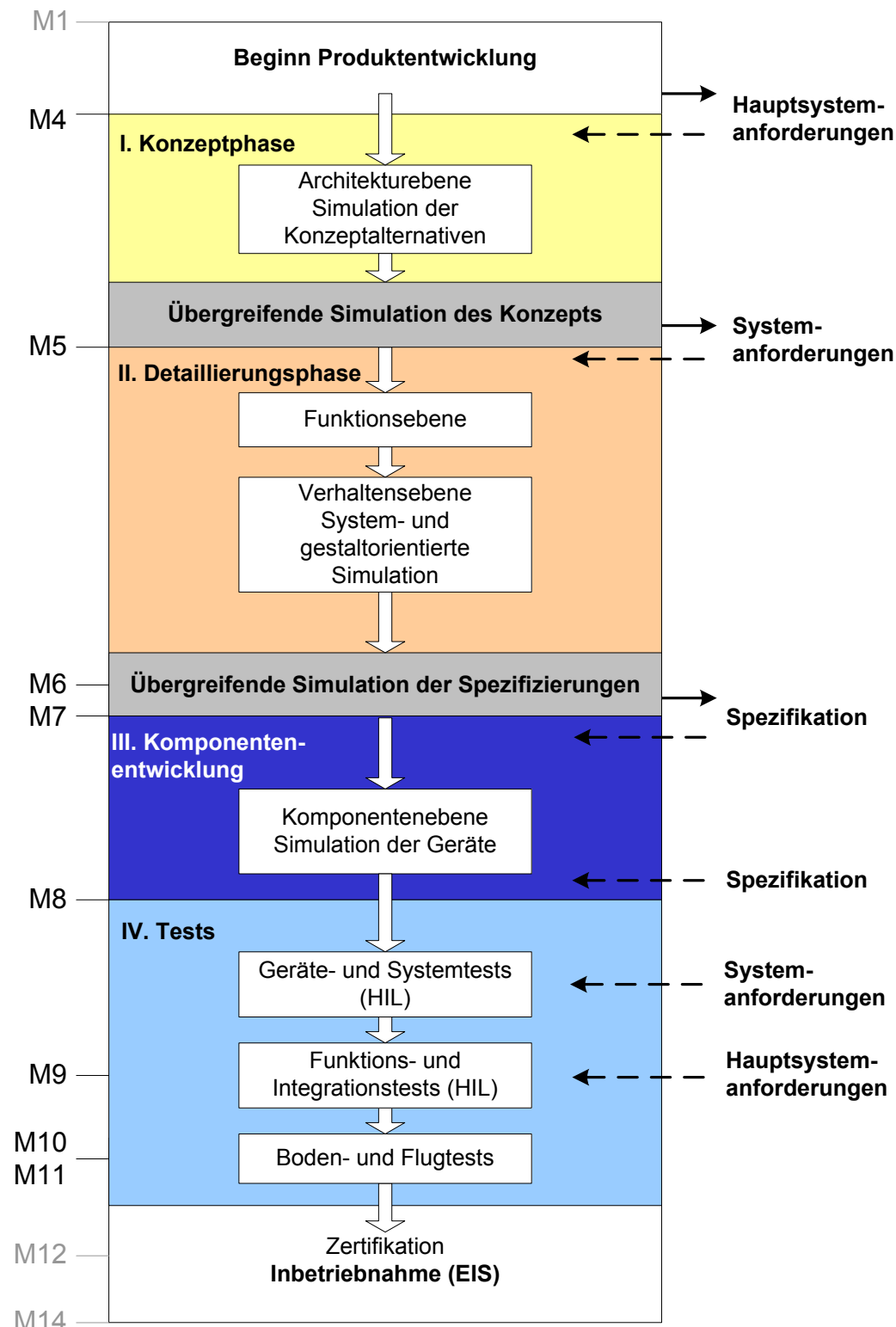


Abbildung 4-7 Systemübergreifender Simulationseinsatz

Die Implementierung der Koordinationsfunktion erfordert ein ganzheitliches Systemverständnis. Für die systemübergreifenden Simulationen müssen die Schnittstellen zwischen den System- bzw. Fachabteilungen bekannt sein. Zusätzlich müssen die Datenflüsse bekannt sein, um den an der Entwicklung beteiligten Mitarbeitern die in der jeweiligen Prozessphase benötigten Informationen zeitnah bereitzustellen und gleichzeitig eine Datenüberflutung zu vermeiden. Aufgrund ihrer interdisziplinären Fähigkeiten sind Systemingenieure für die Implementierung der Koordinationsfunktion im Unternehmen besonders geeignet. Unter dem

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

Begriff des „Systems Engineering“ ist ein interdisziplinärer Ansatz zur Überprüfung der gestellten Anforderungen unter Berücksichtigung des Gesamtproblems (Kosten, Betrieb, Zeit, Test, Erstellung, Planung usw.) von der Konzeption über die Produktion bis hin zum Betrieb zu verstehen [70]. Zusätzlich zu den Systemingenieuren unterstützen Mitglieder der an der Entwicklung beteiligten Fachabteilungen die Implementierung, um so zusätzliches Fachwissen bereitzustellen.

Die multidisziplinäre Simulation muss mit allen Systemen durchgeführt werden, die enge Wechselwirkungen aufweisen. Um den Modellierungsaufwand allerdings beherrschen zu können und die Simulationszeit bei hoher Modelldetaillierung in angemessenen Grenzen zu halten, müssen Systemgrenzen für die Simulation gezogen werden. Die Systemumgebung der multidisziplinären Simulation kann z. B. mit Randbedingungen (Parametern) oder Umgebungsmodellen abgebildet werden. Ansonsten würden für die systemübergreifenden Simulationen immer alle Flugzeugsysteme simuliert werden, und dieses hätte einen enormen Einsatz von Ressourcen zur Folge. Es ist auch möglich, Modelle verschiedener Modellabstraktionsebenen zu kombinieren. Dieses Vorgehen ergibt immer dann Sinn, wenn das Verhalten eines bestimmten Teilsystems genauer untersucht werden soll und die Interaktionen mit angrenzenden Teilsystemen berücksichtigt werden sollen, deren Verhalten aber eine untergeordnete Rolle spielt. Der Vorteil dieses Vorgehens besteht in einer Reduzierung der Simulationszeit.

Ein weiterer wichtiger Punkt besteht darin, dass Modellen und Simulationsergebnissen nicht blind vertraut werden darf, ohne diese einer gründlichen Validierung und Verifikation unterzogen zu haben. Oftmals ist dies nur mithilfe von Messergebnissen realer Versuche möglich. Aus diesem Grund sind reale Tests aus dem Entwicklungsprozess nie ganz wegzudenken. Für die Verifikation und Validierung von Simulationsmodellen bietet sich der VV&A-Prozess (engl.: Verification, Validation and Accreditation) an [117]. Ziel dieses Prozesses ist es zu gewährleisten, dass ein Modell den Anforderungen entspricht (Verifikation) und dass das Modell zusätzlich das Verhalten des realen Systems in bestimmten Grenzen widerspiegelt (Validierung). Zusätzlich wird geprüft, ob ein Modell überhaupt für den Verwendungszweck geeignet ist (Akkreditierung). Der VV&A-Prozess beantwortet damit die folgenden drei Fragen:

- Verifikation: Wurde das Modell korrekt erstellt?
- Validierung: Entspricht das Modell der Wirklichkeit?
- Zulassung: Erfüllt es die Bedürfnisse?

Für weitere Informationen zu diesem Prozess sei an dieser Stelle auf folgende Literatur verwiesen [117], [144]. Die Validierung und Verifikation der Modelle ist wie bisher Aufgabe der Fachabteilungen, allerdings erfolgt zusätzlich eine Kontrolle der Modelle, die in der Koordinationsfunktion implementiert ist. Im folgenden Abschnitt wird untersucht, welche der in Kapitel 2.3.2 beschriebenen Simulationsvarianten sich für die systemübergreifenden Simulationen im Entwicklungsprozess eignen.

Die Verwendung eines multidisziplinären Simulationsprogramms für die systemübergreifende Simulation hat den Nachteil, dass es zu Doppeltätigkeiten kommen kann. In den Prozessphasen vor und zwischen den systemübergreifenden Simulationen werden alleinstehende Simulationen eingesetzt, d. h., es werden die am besten geeigneten Programme der jeweiligen Domäne eingesetzt. Aus diesem Grund wird ein Großteil der erstellten Modelle nicht mit der Modellierung der für die systemübergreifende Simulation eingesetzten Software übereinstimmen. Dies hat zur Folge, dass die Modelle zusätzlich konvertiert werden müssen, woraus ein erheblicher Aufwand resultiert. Zudem wird die Flexibilität bzw. die Funktionsvielfalt in der Simulation stark eingeschränkt, da nur ein einziges multidisziplinäres Simulationsprogramm verwendet wird. Diese Variante scheidet damit für das vorgestellte Konzept aus. Für eine systemübergreifende Simulation sind aufgrund der Nutzung vorhandener Modelle aus den Fachabteilungen die zweite und dritte Variante besser geeignet (siehe Abbildung 2-8).

Die getrennte Modellierung und anschließende gemeinsame Simulation in einem Simulationsprogramm ermöglichen es, die Modelle getrennt in unterschiedlichen Modellierungswerkzeugen zu erstellen bzw. vorhandene Modelle wiederzuverwenden, um diese in ein einziges multidisziplinäres Programm zu importieren und eine Gesamtsimulation durchzuführen. Voraussetzung für diese Möglichkeit ist eine Modellschnittstelle. Ein Großteil der kommerziellen Softwareprogramme besitzt eine C-Code-Schnittstelle, die es ermöglicht,

4.1 Simulationskonzept

Modelle in C-Code umzuwandeln oder C-Code in das Modell zu importieren. Diese Variante ermöglicht die Optimierung der Teilmodelle der Fachabteilungen in alleinstehenden Simulationen, sodass sich bei der Fehlersuche der übergreifenden Simulation auf die Systemschnittstellen und -interaktionen konzentriert werden kann. In Abbildung 4-8 ist die Spezifikation eines C-Codes für den Programmanfang dargestellt. In diesem auskommentierten Codekopf sind Informationen über die Datei, die Ein- und Ausgangsgrößen, die Parameter, das Erstellungsdatum und die Version sowie Kontaktdaten des Erstellers enthalten.

```

/*
File:                name of this file

Version:            unique number of version or issue

Date of first issue: dd/mm/yyyy

Date of current issue: dd/mm/yyyy

Built with:         language of this file, tool (software,
                    version) used creating this file

Author:            name of person/group in charge of
                    development

Contact:           name of person to be contacted in case
                    of questions

Telephone:         phone number of contact

Email:            Email of contact

Special Requirements: other requirements not yet mentioned
                    for the use of this file, e.g. function
                    calls that are required in a certain
                    directory or whether this file has to
                    be stored at a specific path etc.

Copyright (c):     if applicable

Changes from last issue: list of changes

Purpose:           scope of this code/program

Description:       short description of function(s)
provided by this file

Integration:       short description of how and where to
                    insert this file into the simulation
                    environment

Library Calls:     list of all library calls made from
                    this file

Constants:         list of all used constants:
                    <Name>      <Type>      <description>

Variables:         list of all used variables and their
                    dimensions:
                    <Name (dimension)> <Type>  <description>
*/

```

Abbildung 4-8 Spezifikationsbeispiel eines C-Code-Anfangs

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

Die in der Datei enthaltenen Informationen unterstützen die Wiederverwendung der Modelle und verkürzen zusätzlich den Aufwand bei der Weiterverwendung.

Eine weitere Möglichkeit zur Realisierung der systemübergreifenden Simulation stellt die Simulatorkopplung dar. Sie bietet die Möglichkeit, vorhandene Modelle in unterschiedlichen Programmen zu modellieren bzw. bestehende Modelle zu nutzen und in unterschiedlichen gekoppelten Simulationsprogrammen zu simulieren. Im Gegensatz zum Modellimport können in der festen Simulationskopplung (Ko- bzw. verteilte Simulation) Modelle schnell angepasst werden, da die jeweiligen Simulationsprogramme direkt an der Simulation beteiligt sind und die Modelle direkt im Erstellungsprogramm geändert werden können. Hierdurch vereinfacht sich die Fehlersuche im Modell erheblich, da keine umständlichen Konvertierungsschritte nötig sind. Zum einen ermöglicht die Simulationskopplung die Nutzung bestehender Modelle, zum anderen kann die Programmvierfalt beibehalten werden und damit das am besten für die jeweilige Simulation geeignete Programm eingesetzt werden. Zur Kopplung der Simulatoren sind geeignete Schnittstellen zwischen den Programmen erforderlich bzw. ist eine Simulationsumgebung für eine verteilte Simulation zu erstellen. An dieser Stelle sei darauf hingewiesen, dass nur wenige Programme eine Kosimulationsschnittstelle zu anderen Programmen anbieten (z. B. Saber und Matlab/Simulink). Umsetzungen einer losen Kopplung und einer verteilten Simulation mit der HLA werden in Kapitel 5.2.4 beschrieben. Für die Variante 2 bzw. 3 müssen Richtlinien für Modellschnittstellen und Modellaustauschformate von der Koordinationsfunktion definiert werden. Zusätzlich muss sie die Einhaltung dieser Richtlinien überwachen. In Kapitel 5.3 wird aus diesem Grund ein standardisiertes Austauschformat für die lose Kopplung von Simulationsprogrammen spezifiziert.

Es wird deutlich, dass für die systemübergreifende Simulation Modellschnittstellen eine übergeordnete Rolle spielen. Aus diesem Grund werden im Folgenden Möglichkeiten für die Synchronisation von Modellschnittstellen vorgestellt. Für Simulationen, an denen mehrere Personen beteiligt sind, müssen Schnittstellendokumente erstellt werden. Durch den Einsatz dieser standardisierten Dokumentvorlagen werden der Modellaustausch sowie die Integration der Modelle auf anderen Simulationsplattformen vereinfacht. In diesen Dokumenten müssen Informationen über Modellversion, Modellnamen, Modellersteller, Datum usw. enthalten sein. Die Dokumentation der Ein- und Ausgangsvariablen ist für die systemübergreifende Simulation von besonderer Bedeutung, da sie die physikalischen Interaktionen zwischen den Modellen beschreibt. In Tabelle 4-2 ist ein Überblick über die wichtigsten Eigenschaften dieser Variablen dargestellt.

Name	Typ	Beschreibung	Min.	Max.	Letzte Änderung	Einheit
O2_in	double	Sauerstofffluss in das Modell	0	10	Version 2	kg/s
...						

Tabelle 4-2 Funktionale Variableneigenschaften

Für eine reibungslose Modellintegration sind Informationen über den Variablentyp (z. B. Integer, Real, Character usw.), die Modellgrenzen im Bezug auf die Variable sowie die Einheit der Variablen erforderlich.

Für die einfache Modellerweiterung ohne eine Schnittstellenanpassung ist es sinnvoll, Modelle zu kapseln, d. h., die Schnittstellen vom eigentlichen Modell zu trennen. Die Schnittstellen werden in einer Schnittstellenarchitektur, die das Modell umschließt, festgelegt. Es sollte darauf geachtet werden, dass zusätzliche Eingänge für weitere Modellvariablen feilgehalten werden. Die Schnittstellenarchitektur wird einmal festgelegt, sodass angrenzende Modelle (bzw. angrenzende Schnittstellenarchitekturen) immer die gleichen Schnittstellen aufweisen. Diese Technik ermöglicht die kontinuierliche Weiterentwicklung des Modellverhaltens ohne Beeinflussung der Modellschnittstellen. In Abbildung 4-9 ist diese Technik, die in ähnlicher Form auch in der Modellierungssprache VHDL (engl.: Very High Speed Integrated Circuit Hardware Description Language) verwendet wird, dargestellt [29].

4.1 Simulationskonzept

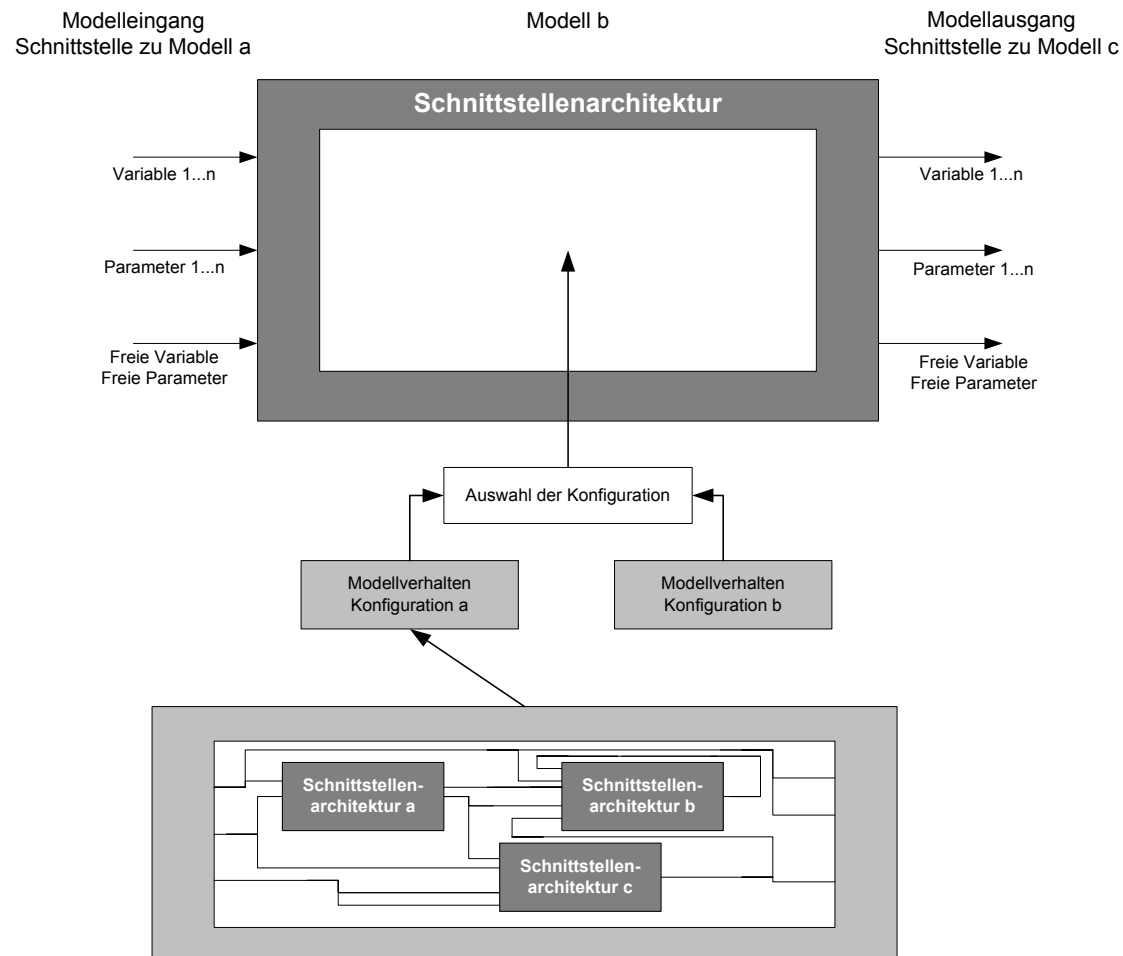


Abbildung 4-9 Kapselung von Modellen

Ist eine zusätzliche Variable für das Modellverhalten erforderlich, steht diese als freie Variable bereits in der Schnittstelle bereit und muss lediglich umbenannt werden. Das in der Schnittstellenarchitektur gekapselte Modellverhalten kann hierarchisch aufgebaut sein und wiederum weitere Schnittstellenarchitekturen enthalten. Zusätzlich können verschiedene Modellkonfigurationen in die Schnittstellenarchitektur integriert werden. Diese Technik ermöglicht es, das Modellverhalten kontinuierlich zu verändern, ohne dabei eine Schnittstellenanpassung durchführen zu müssen. In Kapitel 5.2.1 wird zu diesem Thema eine erweiterbare Matlab/Simulink-Schnittstelle vorgestellt.

Eine weitere Möglichkeit zur Reduzierung des Modellierungsaufwands bietet die Modularisierung. Unter dem Begriff der Modularisierung versteht man das Aufteilen eines Gesamtsystems in Teilsysteme, die als Modul bezeichnet werden. In der Modellierung bedeutet dies, dass ein Modell aus standardisierten Einzelmodellen zusammengesetzt wird. Die Einzelmodelle (Module) lassen sich dabei unterschiedlich zu einem Gesamtmodell zusammensetzen. Module werden dabei hauptsächlich verwendet, um sie leicht gegen andere Module austauschen zu können oder neue Module hinzuzufügen. Um Probleme der Kompatibilität gering zu halten, wird für die Module eine einheitliche Schnittstelle benötigt. In den Simulationsprogrammen werden dem Benutzer aus diesem Grund Modellbibliotheken mit vorgefertigten Komponenten verschiedenster physikalischer Fachdisziplinen zur Verfügung gestellt [11], [21]. Zusätzlich zu diesen Bibliotheken ist es wichtig, dass die einzelnen Entwicklungsabteilungen eigene Modellbibliotheken aufbauen. Für den Aufbau, die Wartung und Weiterentwicklung einer solchen Bibliothek werden zusätzliche Ressourcen gebunden, die sich aber durch die einfache Wiederverwendung der Module innerhalb kürzester Zeit amortisieren [112]. In Kapitel 5.2.1 wird der Modellaufbau eines Brennstoffzellensystems mithilfe einer benutzerdefinierten Modellbibliothek dargestellt. Im folgendem Abschnitt werden die Auswirkungen eines durchgängigen Simulationskonzepts auf die Entwicklungskosten und -zeit beschrieben.

4.1.2.3 Ressourceneinsatz des Simulationskonzeptes

Die Einführung eines neuen Konzeptes, einer neuen Software usw. ist zunächst immer mit einem Anstieg der benötigten Ressourcen verbunden. Allgemein ist festzustellen, dass die Investitionskosten nicht nur aus den Kosten für die Anschaffung der Hard- bzw. Software gebildet werden. Vielmehr werden sie im Wesentlichen aus den Kosten für Personalfortbildungen und damit einhergehende Kapazitätsausfälle sowie innerbetriebliche Strukturmaßnahmen zur Integration des Konzeptes in bestehende Betriebsabläufe gebildet [59]. In Abbildung 4-10 ist dargestellt, wie sich die Produktivität als Funktion der Zeit für eine Entwicklung mit und ohne Softwareprogramm verhält.

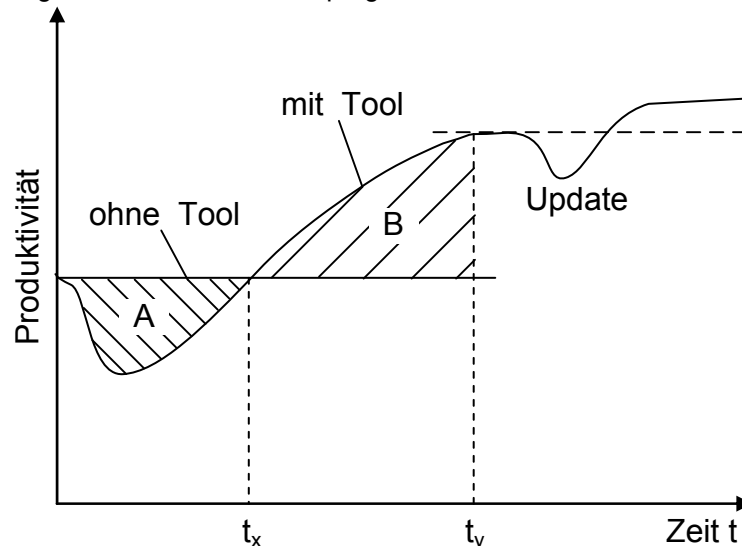


Abbildung 4-10 Produktivitätsänderung bei der Einführung eines Simulationsprogramms [59]

Direkt nach dem Kauf einer Software ist zunächst ein Zeitpuffer für die Installation vorgesehen. Während der Einführung sinkt die Produktion zunächst. Nach dem Durchschreiten eines Produktivitätsminimums wird zum Zeitpunkt t_x die alte Produktivität erreicht. Die Wirtschaftlichkeit der Software wird zu dem Zeitpunkt t_y erreicht, wenn das aufgelaufene Defizit der Produktion (Fläche A) kompensiert wurde (Fläche $B \geq A$). Dieses Verhalten gilt ebenfalls für ein neues Simulationskonzept. Der Mehraufwand bei der Einführung wird sich erst nach einer gewissen Zeit amortisieren. Zusätzlich führt der frühe und kontinuierliche Einsatz von Simulation in der Entwicklung zu einem Anstieg der Ressourcenbindung in den frühen Prozessphasen, allerdings werden Fehlentwicklungen früher erkannt (vgl. Abbildung 4-6), wodurch sich insgesamt eine Reduktion der Entwicklungskosten ergibt. Zudem steigt durch die Verlagerung der Entwicklungsaktivitäten auf die frühen Prozessphasen die Entscheidungsqualität.

Im derzeitigen Entwicklungsprozess werden voneinander unabhängige Simulationen in den einzelnen Fachabteilungen durchgeführt. Für die Simulationen können so immer die situationsbedingt am besten geeigneten Programme eingesetzt werden. Das vorgestellte Simulationskonzept stellt eine Kombination aus alleinstehenden und systemübergreifenden Simulationen dar. Damit wird gewährleistet, dass auch mit dem neuen Simulationskonzept die Programmvierfalt erhalten bleibt. Die an der Entwicklung beteiligten Mitarbeiter können weiterhin die bereits eingesetzten Programme benutzen, wodurch zusätzliche Schulungsmaßnahmen ausbleiben. Allerdings müssen für die systemübergreifenden Simulationen die Modelle von den Fachabteilungen nach den von der Koordinationsfunktion definierten Richtlinien angepasst werden. Zusätzliche Ressourcen werden für die Kontrolle der Modelle, die Abstimmung der Schnittstellen und die Durchführung der Simulation gebunden. Der Umfang des Aufwands ist dabei stark von der eingesetzten Variante für die domänenübergreifende Simulation abhängig. Unabhängig davon wird dieser zusätzliche Aufwand durch die bereits beschriebenen Effekte kompensiert.

4.1.3 Zusammenfassung des Simulationskonzeptes

In diesem Kapitel werden die wichtigsten Punkte des Simulationskonzeptes zusammengefasst und die Anforderungen an das Konzept verifiziert. Aus diesem Grund wird zunächst

4.1 Simulationskonzept

überprüft, ob die im Kapitel 3 aufgestellten Anforderungen an die Simulation im Prozess erfüllt werden.

Durch die Einführung von Richtlinien und Standards für die Modellschnittstellen und den Modellaustausch wird die Wiederverwendung der Modelle ermöglicht. Dabei wird die Einhaltung der Richtlinien kontrolliert. Dies hat zur Folge, dass die Testvorrichtung schon während der Detaillierungsphase erstellt werden kann und somit zur Erfüllung der Anforderungen zur Parallelisierung von Aktivitäten beiträgt.

Die frühzeitige Überprüfung von Systemwechselwirkungen wird mithilfe der systemübergreifenden Simulationen nach der Konzeptphase und nach der Detaillierungsphase realisiert (vgl. Kapitel 4.1.2.2). Durch den Einsatz der Simulationen wird zusätzlich die Anforderung einer virtuellen Systemüberprüfung erfüllt. Zudem wird durch die Kombination aus alleinstehenden und systemübergreifenden Simulationen die Programmvierfalt gewährleistet.

Der Einsatz von Simulationen in den frühen Entwicklungsphasen unterstützt die Konzeptauswahl und erhöht die Entscheidungsqualität. Das vorgestellte Simulationskonzept schreibt den kontinuierlichen Einsatz von Simulationen von der Konzeptphase bis zum Ende der Testphase vor. Ein Modellierungskonzept mit verschiedenen Detaillierungsebenen (vgl. Abbildung 2-7) ermöglicht die Anpassung der Modelle an den jeweiligen Entwicklungsstand und unterstützt damit Anforderung eines durchgängigen Simulationseinsatzes.

Eine vollständige Darstellung zur Umsetzung der Anforderungen im vorgestellten Simulationskonzept ist in Tabelle 4-3 enthalten.

Zusammenfassend lässt sich sagen, dass die Vorteile der systemübergreifenden Simulation in der frühen virtuellen Verifizierung der Entwicklung liegen. Inkompatibilitäten bzw. Entwicklungsfehler werden in den frühen Prozessphasen und nicht erst in der Testphase erkannt. Änderungen können damit in kürzeren Änderungsschleifen mit geringeren Kosten behoben werden, als dieses mit realen Tests möglich ist. Eine Schwierigkeit des Simulationseinsatzes in den frühen Entwicklungsphasen besteht darin, dass zu diesem Zeitpunkt wenige Daten für den Aufbau der Simulationsmodelle existieren. Für die Systemsimulation stellt dies kein Problem dar, da zunächst mit Annahmen oder Datenmaterial aus vorangegangenen Entwicklungen gearbeitet werden kann. Für gestaltorientierte Simulationen (siehe Kapitel 7.2) ergibt sich daraus sehr wohl ein Problem, da diese Simulationen auf Geometriedaten basieren, die in den frühen Entwicklungsphasen noch nicht vorhanden sind. Dieses Problem wird eingehend in Kapitel 5.2.3 beschrieben.

Bedingt durch den frühzeitigen Simulationseinsatz, erfolgt eine Vorverlegung der Verifikation in den linken Strang des V-Modells. Die systemübergreifenden Simulationen ermöglichen es, dass eine Vielzahl von Szenarien, zum Beispiel durch automatische Generierung von Störfällen, getestet werden, die ohne Simulation eventuell gar nicht in den realen Tests betrachtet worden wären. Damit steigt bei der Durchführung von übergreifenden Simulationen die Entwicklungsqualität an.

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

Anforderung: Simulation im Prozess		Umsetzung
Parallelisierung von Aktivitäten	Modellwiederverwendung	Weitergabe der Modelle aus der Detaillierungsphase
	Frühzeitige Erstellung der Testvorrichtung	Modellbasierter Aufbau der Testvorrichtung (M7)
	Einheitliches globales Umgebungsmodell	Definition durch die Koordinationsfunktion
Frühzeitige Überprüfung von Systemwechselwirkungen	Konsistente Modellierung und Simulation in der Entwicklung	Schnittstellenstandard
	Systemübergreifende Simulation gewährleisten	Multidisziplinäre Simulationsvarianten 2 und 3
	Schnittstellenstandards	Definition durch das Koordinationsteam (z. B. ICD)
	Standardisierter Modellaustausch	Definition durch die Koordinationsfunktion
	Virtuelle Überprüfung der Systemwechselwirkungen	Systemübergreifende Simulation (M5–M7)
	Systemübergreifende Entwicklungs-koordination	Koordinationsfunktion
	Einfache Modellintegration	Modularisierung Schnittstellendefinition
	Reduzierung der Hardwaretests und Prototypen	Einsatz von Simulation in der Testphase
	Einsatz des am besten geeigneten Simulationsprogramms	Kombination alleinstehender und übergreifender Simulation
Durchgängiger Simulationseinsatz	Validierung der Modelle	In den Fachabteilungen und durch die Koordinationsfunktion
	Simulationsgestützte Konzeptauswahl	Systemübergreifende Simulation zu M5
	Steigerung des Innovationsgrades	Einsatz von Simulation in den frühen Prozessphasen
	Erstellung von Konzeptalternativen	Wiederverwendung von Modellen
	Lieferung von Komponentenmodellen durch Zulieferer	Zulieferer verpflichtet sich zur Lieferung eines Konzeptmodells
	Kontinuierlicher Simulationseinsatz	Simulationseinsatz in allen Entwicklungsphasen
	Modularisierung der Simulationsmodelle	Fachdisziplinspezifische Modellbibliotheken
	Reduktion des Reparaturverhaltens	Früherkennung von Fehlentwicklungen
	Hierarchisches Modellierungskonzept	Modellabstraktionsgrade

Tabelle 4-3 Umsetzung der Anforderungen im Simulationskonzept

In diesem Kapitel wurde deutlich, dass für ein Simulationskonzept weitere Voraussetzungen nötig sind. So können der Modellaustausch sowie eine Verbesserung der Kommunikation zwischen Teilsystementwicklungen nur durch den Einsatz eines geeigneten Informationsmanagementsystems erreicht werden. Zusätzlich zu den im Kapitel 4.1 vorgestellten Methoden für die Schnittstellensynchronisation werden durch eine standardisierte Modellierungssprache der Modellaustausch sowie die Kommunikation erheblich verbessert und somit auch die Schnittstellenprobleme deutlich verringert. Aus diesen Gründen werden in den folgenden Kapiteln zunächst ein Konzept eines Informationsmanagementsystems und anschließend eine Konzept für die standardisierte Modellierung vorgestellt.

4.2 Informationsmanagement

Die Bereitstellung von Informationen ist für eine erfolgreiche Produktentwicklung von zentraler Bedeutung. Nach Gebhard sind für eine effiziente Produktentwicklung zwei Voraussetzungen entscheidend. Zum einen ist ein hoher Parallelisierungsgrad der bisher sequenziellen Entwicklungsphasen erforderlich, zum anderen wird eine allgemeine Datenbasis benötigt, die für alle an der Entwicklung Beteiligten jederzeit verfügbar ist [55]. Der Erfüllung des zweiten Punktes ist dieses Kapitel gewidmet. Aus diesem Grund werden zunächst verschiedene Konzeptalternativen eines Informationsmanagements bewertet, und anschließend wird das am besten geeignete Informationsmanagementkonzept detailliert.

4.2.1 Alternativen für das Informationsmanagement

Hinsichtlich des Informationsmanagements werden analog zur Simulationskonzeptauswahl drei Konzepte vorgestellt, die den Entwicklungsprozess und den Einsatz von Simulation im Prozess ermöglichen und unterstützen sollen.

Konzeptalternative 1: Koordinationsteam

Der Informationsaustausch zwischen den an der Entwicklung beteiligten Mitarbeitern erfolgt wie bisher per E-Mail, Telefon und persönlichen Absprachen. Zusätzlich stehen alle derzeit eingesetzten Informationsquellen, z. B. Netzlaufwerke, lokale Datenablage, mobile Speichermedien, Datenbanken usw., zur Verfügung. Um den Informationsaustausch zu regeln, wird ein Koordinationsteam eingesetzt, das den Datenaustausch organisiert und für die Abstimmung zwischen den unterschiedlichen Fachabteilungen sorgt. Das Koordinationsteam stellt in diesem Konzept eine Umsetzung der in Kapitel 4.1.2.2 beschriebenen Koordinationsfunktion dar. Es verschickt Benachrichtigungen an die Entwicklungsabteilungen, um über aktualisierte Daten usw. zu informieren und den Fortschritt der einzelnen Entwicklungsabteilungen zu kommunizieren. Hierfür ist ein direkter Informationsaustausch zwischen dem Team und den einzelnen Fachabteilungen nötig und es ist darauf zu achten, dass das Koordinationsteam nur die Informationen verschickt bzw. freigibt, die von der entsprechenden Abteilung benötigt werden. Eine weitere Aufgabe dieses Teams besteht in der Vergabe von Zugriffsrechten für z. B. Netzlaufwerke und Datenbanken. In Abbildung 4-11 ist dieser Informationsaustausch dargestellt.

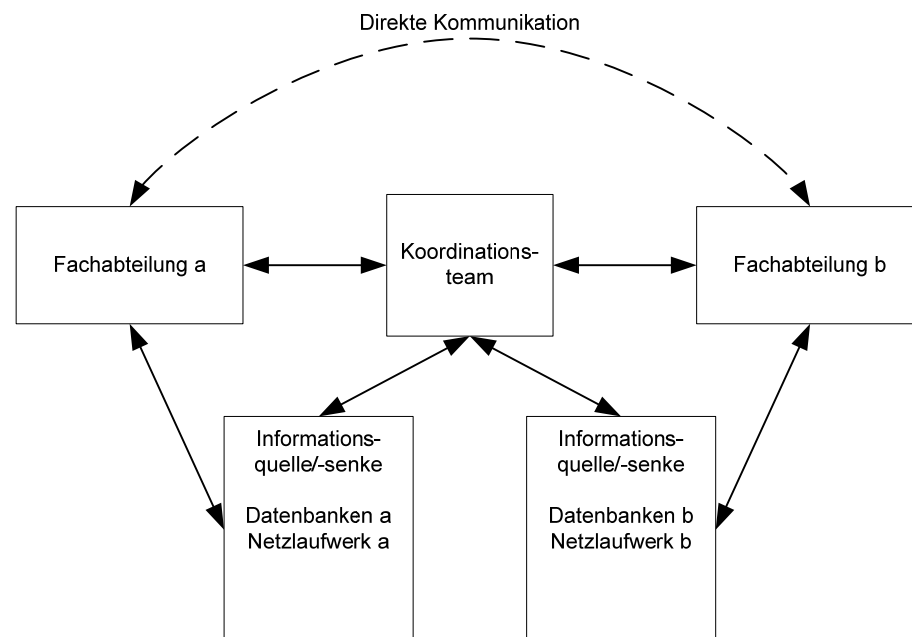


Abbildung 4-11 Informationsbereitstellung mit Koordinationsteam

Ein Vorteil dieses Konzepts ist die Weiterverwendung bestehender Infrastrukturen, somit entfallen Anschaffungskosten für Hardware und Software. Zusätzlich wird die systemübergreifende Abstimmung koordiniert und unterstützt. Allerdings ist ein intensiver Informationsaustausch zwischen dem Koordinationsteam und den einzelnen Fachabteilungen erforderlich,

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

der einen erheblichen Ressourceneinsatz erfordert. Bedingt durch den Einsatz vieler verschiedener Informationsquellen, erhöht sich der Beschaffungsaufwand für die Fachabteilungen erheblich, wodurch sich die eigentliche wertschöpfende Tätigkeit stark reduziert [108]. Aus dieser Infrastruktur ergibt sich eine dezentrale Rechtevergabe, d. h., für jede Informationsquelle müssen Zugriffsrechte vergeben werden. Zudem müssen die Informationsquellen überwacht werden, um die Fachabteilungen über aktuelle Informationen auf dem Laufenden zu halten. Dies erfordert einen hohen Aufwand seitens des Koordinationsteams, sodass die Gefahr einer Überlastung besteht und die Fachabteilungen entweder keine oder nur verspätet Informationen erhalten oder mit Informationen geradezu überschüttet werden.

Im Vergleich zum derzeitigen Vorgehen kann die Entwicklungsabteilung Informationen direkt über das Koordinationsteam beziehen. Durch die hohe Auslastung dieses Teams wird allerdings die direkte Kommunikation (z. B. E-Mail, Telefon, Datenaustausch per mobilen Speicher, usw.) zwischen den Fachabteilungen begünstigt. Die ausgetauschten Informationen sind somit für das Koordinationsteam nicht sichtbar und können damit nicht an weitere Fachabteilungen, für die diese Informationen ebenfalls von Interesse sind, weitergegeben werden. Ein weiterer Nachteil ist, dass der einzelne Entwickler keinen Überblick über die verfügbaren Informationen hat und daher stark vom Koordinationsteam abhängig ist.

Konzeptalternative 2: zentrale Datenbank

In einer zentralen Datenbank, auf die alle an der Entwicklung beteiligten Abteilungen zugreifen können, werden alle Entwicklungsdaten (z. B. Dokumente und Simulationsergebnisse) gespeichert. Diese Datenbank fungiert als einzige Datenaustauschplattform im Entwicklungsprozess, d. h., die zentrale Datenbank ersetzt bestehende Datenmanagementsysteme. Die Kontrolle des Datenflusses (z. B. die Ablage von Daten), die Vergabe der Zugriffsrechte und die Bereitstellung der benötigten Infrastruktur erfolgen mittels einer im Unternehmen implementierten Koordinationsfunktion. Die an der Entwicklung beteiligten Mitarbeiter können Daten entsprechend der individuellen Zugriffsrechte in der Datenbank ablegen und entnehmen. Bei der Einstellung neuer Daten wird eine automatische Benachrichtigung an die Nutzer gesendet, für die diese Daten von Interesse sind. In Abbildung 4-12 ist der Aufbau einer zentralen Datenbank dargestellt.

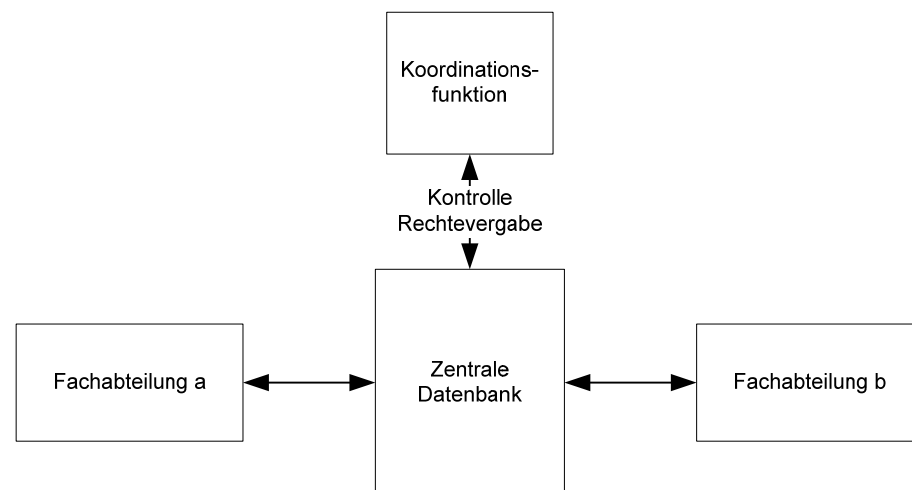


Abbildung 4-12 Informationsbereitstellung mit zentraler Datenbank

Der gesamte Informationsfluss läuft über die zentrale Datenbank, die von der Koordinationsfunktion überwacht wird. Ein Vorteil dieses Konzeptes liegt im geringen Betreuungsaufwand der Datenbank sowie der zentralen Vergabe der Zugriffsrechte, wodurch gewährleistet wird, dass der Benutzer nur die für ihn relevanten Entwicklungsdaten einsehen kann. Zusätzlich wird die Datensuche erheblich vereinfacht. Ein weiterer Vorteil besteht darin, dass sich der Nutzer an nur einer Datenbank anmelden muss, um auf alle Daten zugreifen zu können. Für die Beschaffung sowie die Einstellung von Daten wird lediglich ein Softwareprogramm benötigt, wodurch sich die Benutzung zusätzlich vereinfacht.

Ein Nachteil dieses Konzepts besteht darin, dass nicht auf bestehende Infrastrukturen zurückgegriffen werden kann. Die bestehende Infrastruktur für den Informationsaustausch wird durch eine zentrale Datenbank ersetzt, daraus resultieren enorme Investitionskosten. Zudem besteht keine Redundanz, d. h., im Falle eines Datenbankausfalles kann es zum

4.2 Informationsmanagement

Erliegen der gesamten Entwicklung kommen. Die Kosten für ein zweites Sicherungssystem (engl.: backup) sind hoch, da dies ebenfalls alle Entwicklungsdaten enthalten muss. Wird wie im ersten Konzept auf eine Infrastruktur mit vielen verschiedenen Informationsquellen aufgebaut, ist dieses nicht der Fall, da der Ausfall eines einzelnen Systems nicht den gesamten Informationsfluss unterbricht und damit den Entwicklungsprozess nicht zum Erliegen bringt. Ein weiterer Nachteil besteht darin, dass Daten sehr unterschiedliche Strukturen aufweisen. Eine Datenbank für Systemsimulatoren weist andere Eigenschaften auf als eine Datenbank für CFD-Simulationen [122]. Somit kann nicht gewährleistet werden, dass die am besten geeignete Datenbanktechnik eingesetzt wird. Die Umstellung auf dieses Konzept erfordert, dass die bereits vorhandenen Daten in ein für die zentrale Datenbank verständliches Format konvertiert werden, dies erfordert einen zusätzlichen Einsatz von Ressourcen.

Konzept 3: zentrale Informationsbasis

Den Kern dieses Konzeptes bildet eine zentrale Informationsbasis (IB), die die Funktion einer zentralen Plattform für den Austausch von Daten und Informationen, ähnlich den in Kapitel 2.4.2 vorgestellten PDM-Systemen, bildet. Der Datenaustausch wird über eine einheitliche Plattform organisiert, und integrierte Systemdienste, wie Konfigurationsmanagement, Rechtevergabe und Datensicherheit werden von einer Koordinationsfunktion bereitgestellt. Die zentrale Informationsbasis basiert auf der im Unternehmen vorhandenen Infrastruktur. Wie bisher kommen die gegenwärtigen Datenbanken und Datenmanagementsysteme zum Einsatz, mit dem Unterschied, dass beim Ablegen der Informationen eine Verknüpfung in der Informationsbasis hinterlegt wird. In der IB werden also lediglich Metadaten gespeichert, die eigentlichen Datensätze liegen weiterhin auf den bisher eingesetzten Informationsquellen. Als Metadaten bezeichnet man allgemein Daten, die Informationen über andere Daten enthalten. Mithilfe der Verknüpfung kann zentral über die Informationsbasis auf die Daten zugegriffen werden. Durch die Rechtevergabe wird festgelegt, auf welche Informationen der einzelne Mitarbeiter zugreifen kann. Zusätzlich werden beim Einstellen der Daten weitere Metadaten, wie z. B. Datum, Ersteller, Version, Abteilung usw., in der IB hinterlegt, um so z. B. die Suchfunktion der IB zu unterstützen. Die Zugriffsrechte dienen ebenfalls zur automatisierten Benachrichtigung der Mitarbeiter. Werden aktuelle Informationen in der IB eingestellt, dann erhalten die Mitarbeiter mit Zugriffsrecht auf diese Informationen eine Benachrichtigung. Es werden also immer die aktuellen Informationen kommuniziert, und gleichzeitig wird eine Datenüberflutung der Mitarbeiter verhindert. In Abbildung 4-13 ist der prinzipielle Aufbau dieses Konzeptes dargestellt.

Ein Vorteil dieses Konzeptes besteht darin, dass die Nutzer sich nur einmal zentral anmelden und dann Zugriff auf alle für sie relevanten Daten haben. Hieraus ergibt sich ebenfalls eine Reduzierung der Verwaltungsaufgaben, so müssen z. B. Zugriffsrechte nur einmal vergeben werden. Gleichzeitig basiert auf den Zugriffsrechten die automatisierte Kommunikation aktueller Daten. Die Investitionskosten sind im Vergleich zum zweiten Konzept niedrig, da auf der bestehenden Infrastruktur aufgebaut wird. Zusätzlich kann der Übergang von der gegenwärtigen Informationsbereitstellung zur zentralen Informationsbasis schrittweise erfolgen, somit können die verschiedenen Informationsquellen nach und nach integriert werden. Der Schulungsaufwand sowie die Einarbeitungszeit sind gering, da nur die Benutzung der Informationsbasis erlernt werden muss. Bedingt durch die in der IB abgelegten Metadaten, vereinfachen sich die Informationsbeschaffung sowie der Umgang mit dem Datenmaterial erheblich. Ein weiterer Vorteil dieses Konzeptes besteht darin, dass immer die am besten geeignete Informationsquelle für die Bereitstellung der Entwicklungsdaten genutzt werden kann.

Ein Nachteil dieses Konzeptes besteht in einem erhöhten Implementierungsaufwand bei der Einführung der Informationsbasis. Des Weiteren besteht die Gefahr, dass nur ein Teil des verfügbaren Datenmaterials in der Informationsbasis hinterlegt wird. Aus diesem Grund muss gewährleistet werden, dass alle Daten über die Informationsbasis eingestellt werden.

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

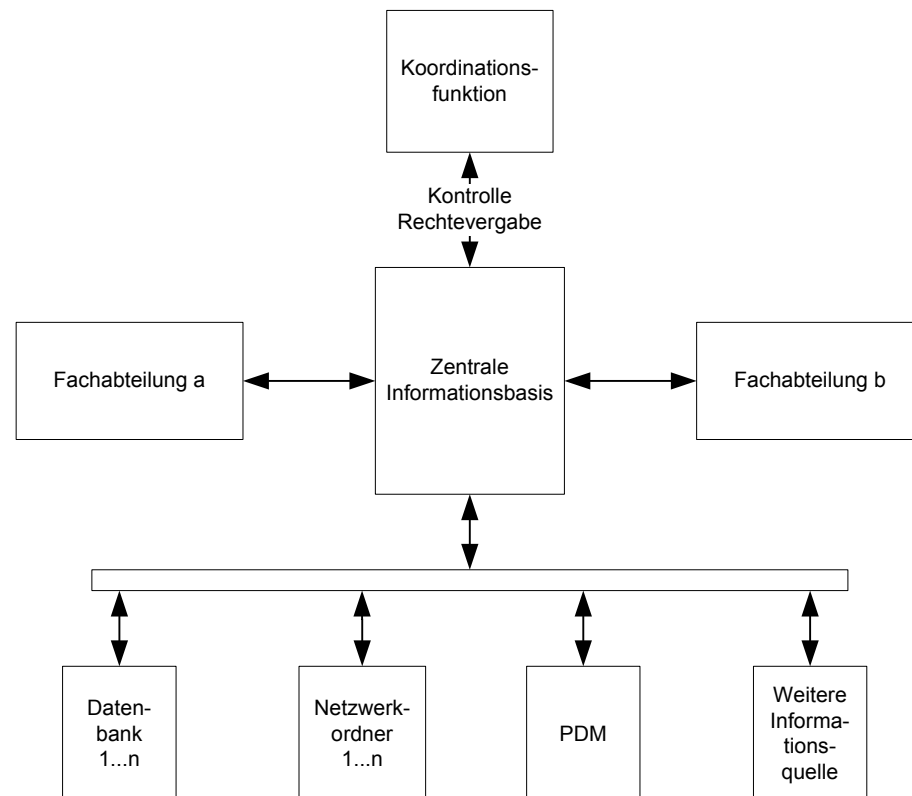


Abbildung 4-13 Informationsbereitstellung mit zentraler Informationsbasis

In der folgenden Tabelle werden verschiedene Kriterien der drei vorgestellten Konzeptalternativen bewertet, die sich an den Anforderungen an das Informationsmanagement aus Kapitel 3 orientieren. Für die Bewertung der Kriterien kommt ein dreistufiges Bewertungsverfahren zum Einsatz. Dabei steht ein „+“ für ein erfülltes, eine „0“ für eine teilweise erfülltes und ein „-“ für ein nicht erfülltes Kriterium.

Kriterien Informationsmanagement	Konzept 1 Koordinationsteam	Konzept 2 Zentrale Datenbank	Konzept 3 Zentrale Informationsbasis
Aufwand der Informationsbeschaffung	0	+	+
Kontinuierliche Kommunikation aktueller Informationen	0	+	+
Betreuungsaufwand	-	+	0
Nutzung der am besten geeigneten Informationsbereitstellung	+	-	+
Nutzung vorhandener Ressourcen	+	-	+
Ressourceneinsatz bei der Einführung	+	-	0

Tabelle 4-4 Bewertung der Grobkonzepte für die Informationsbereitstellung

Das erste Konzept erhält hinsichtlich der Informationsbeschaffung und -kommunikation eine mittlere Bewertung. Diese beiden Punkte hängen direkt vom Koordinationsteam ab, das aufgrund des hohen Betreuungsaufwands oftmals Informationen nicht zeitnah und zum Teil unvollständig zur Verfügung stellt. Hieraus resultiert ein starker Anstieg der direkten Kommunikation zwischen den beteiligten Entwicklern. Diese Form des Informationsaustausches verhindert die Wissensteilung mit Dritten. Für die Steigerung der Entwicklungsqualität ist die Wissensteilung allerdings von entscheidender Bedeutung. An dieser Stelle sei darauf hingewiesen, dass die direkte Kommunikation ein wichtiges Hilfsmittel in der Entwicklung darstellt. Viele Fragen können einfach und schnell per Telefon oder E-Mail gelöst werden. Es muss

4.2 Informationsmanagement

allerdings darauf geachtet werden, dass wenn Daten ausgetauscht werden, die ebenfalls für andere an der Entwicklung Beteiligte relevant sind, diese zentral zur Verfügung gestellt werden. In den Konzepten zwei und drei wird ebenfalls direkte Kommunikation als probates Mittel des Informationsaustausches eingesetzt. Im Vergleich zum ersten Konzept wird diese Form der Kommunikation reduziert, da alle Informationen in der zentralen Datenbank bzw. Informationsbasis zur Verfügung stehen. Ein Beispiel soll dies verdeutlichen. Fachabteilung a benötigt ein Simulationsmodell der Fachabteilung b. Per E-Mail nimmt die Fachabteilung a Kontakt auf. Die Fachabteilung b stellt das benötigte Modell, wenn dieses noch nicht vorhanden ist, in der IB bzw. zentralen Datenbank ein. Automatisch werden alle Nutzer, für die diese Informationen von Interesse sind, benachrichtigt. Zusätzlich hierzu kann Fachabteilung b noch einen Verweis (engl.: link) auf diese Informationen an die Fachabteilung a senden. Im ersten Konzept wird das angeforderte Modell oftmals direkt als E-Mail-Anhang verschickt und ist für Dritte nicht verfügbar. Weitere Nachteile dieses Vorgehens sind eine erhöhte Beanspruchung der Netzwerkressourcen sowie ein Sicherheitsrisiko für den Fall, dass der E-Mail-Verkehr nicht verschlüsselt ist. Ein Vorteil des ersten Konzeptes ist der geringe Einsatz von Ressourcen während der Einführung. Die vorhandenen Kapazitäten zur Datenspeicherung werden weiterhin genutzt, allerdings ist eine für alle Entwickler zugängliche Datenbereitstellung nicht gegeben. Insgesamt erfüllt das erste Konzept die Anforderungen unzureichend.

Die Konzepte zwei und drei legen mit der gemeinsamen Datenbasis den Grundstein zur Intensivierung des Informationstauschs. Beide Konzepte zeichnen sich durch eine einfache Informationsbeschaffung sowie die Kommunikation aktueller Informationen aus. Der kontinuierliche Aufwand für die zentrale Informationsbasis ist allerdings geringfügig höher als für die zentrale Datenbank. Diese Tatsache ist darin begründet, dass in der IB viele verschiedene Informationsquellen eingesetzt werden. Kommen neue Quellen hinzu oder ändern sich alte, macht dies eine Anpassung der IB notwendig. Dieser Nachteil gegenüber dem zweiten Konzept wird durch die geringeren Investitionskosten kompensiert. Zusätzlich stellen unterschiedliche Entwicklungsdaten auch individuelle Anforderungen an die Datenspeicherung. Ein einzelnes Programm kann niemals den Standard eines Programmportfolios erreichen [122], [45]. Einen weiteren Vorteil des ersten und dritten Konzeptes gegenüber der zentralen Datenbank bildet die Informationsaufteilung auf viele unterschiedliche Informationsquellen (z. B. Datenbanken, Netzlaufwerk usw.). Ein Ausfall einer oder auch mehrerer Informationsquellen schränkt die Entwicklung nur geringfügig ein, während der Ausfall der Datenbank in Konzept zwei die Entwicklung zum Erliegen bringt.

Nach Schäppi besteht keine Notwendigkeit, alle Bereiche eines Unternehmens mit einem einzigen System zu unterstützen, da ca. 80 % der Daten in den jeweiligen Bereichen erzeugt, bearbeitet, verwendet und dort gespeichert werden [147]. In diesem Zusammenhang werden allerdings nicht die besonderen Bedürfnisse eines abteilungsübergreifenden Datenaustauschs im systemübergreifenden Entwicklungsprozess berücksichtigt. Dennoch wird deutlich, dass nicht eine Substitution bestehender Systeme zur Datenspeicherung genügt, sondern eine Einbindung des Informationsmanagements in den Prozess nötig ist [188].

Insgesamt erhält das Konzept der zentralen Informationsbasis die höchste Bewertung. Die Entwickler erhalten Benachrichtigungen bei Einstellung neuer Informationen, so dass kontinuierlich die aktuellen Informationen kommuniziert werden, ohne die Empfänger mit Informationen zu überfluten. Der Ressourceneinsatz für die Betreuung der Informationsbasis und für das Einstellen der Informationen ist allerdings abhängig von der Gestaltung der Informationsbasis und der Prozesse, sodass dies in der folgenden Detaillierung besonders zu berücksichtigen ist.

4.2.2 Detaillierung des Informationsmanagements

Ein erfolgreicher Entwicklungsprozess sowie dessen durchgängige Simulationsunterstützung hängen wesentlich von der Datenverfügbarkeit, -konsistenz, den Kommunikationsmöglichkeiten usw. ab. Die Auswertung der Grobkonzeptalternativen im vorangegangenen Kapitel zeigte die Vorteile einer Informationsbasis. Eine Informationsbasis erhöht die Transparenz in der Entwicklung, stellt Kommunikationsmöglichkeiten bereit und nutzt gleichzeitig die vorhandenen Datenbanken und -managementsysteme. Allerdings ist die schlichte Bereitstellung einer Informationsbasis in der Entwicklung nicht ausreichend. Es sind vor allem Vorgehensweisen in der Informationsbereitstellung zu konzipieren, um die Kommunikation der Informationsbasis zu organisieren. In der folgenden Detaillierung des Teilkonzepts soll aufgezeigt werden, welche Informationen zu welchen Zeitpunkten im Prozess bereitgestellt werden

müssen. Weiterhin soll die Organisation der systemübergreifenden Kommunikation gestaltet werden. Das Teilkonzept wird dabei hinsichtlich der Entwicklungsdokumente, der Weiterverwendung von Modellen und Daten sowie der abteilungsübergreifenden Kommunikation detailliert. Ziel des Informationsmanagements ist die Bereitstellung der für eine durchgängige Simulationsunterstützung in der Systementwicklung benötigten Modelle, Daten, Dokumente usw.

4.2.2.1 Bereitstellung der Entwicklungsdokumente

Die Entwicklung eines sicherheitskritischen Produktes stellt hohe Anforderungen an Entwicklungsdokumente. Die strukturierte Ablage der Daten unterstützt die Wiederverwendung sowie die Wissensteilung. Um die Konsistenz der Dokumentationen und der darin enthaltenen Informationen zu gewährleisten, ist es erforderlich festzulegen, welche Dokumentationen erforderlich sind und wann und wie diese Dokumente zu verwenden sind. Im Folgenden werden wichtige Dokumente, die im Rahmen eines Entwicklungsvorhabens zu erstellen sind und für die Simulation von besonderer Bedeutung ist, vorgestellt.

Anforderungsdokumente:

Die Anforderungsdokumente stellen die wichtigsten Dokumente in der Flugzeugsystementwicklung dar. Ausgehend von den Hauptsystemanforderungen werden die Anforderungen mit voranschreitender Entwicklung zunächst in die Systemanforderungen und anschließend in die Systemkomponentenanforderungen bzw. Gerätespezifikationen verfeinert und in Form von Dokumenten abgelegt (vgl. Abbildung 3-3). Bei dieser Dekomposition ist darauf zu achten, dass die jeweils übergeordneten Anforderungen von den Anforderungen der untergeordneten Ebene erfüllt werden. Es muss also zu jedem Zeitpunkt der Entwicklung im linken Strang des V-Modells überprüft werden, ob der jeweilige Entwicklungsschritt zur Erfüllung der übergeordneten Anforderungen beiträgt. Dieses Vorgehen wird als Validierung bezeichnet. Für die Validierung ist die Nachvollziehbarkeit der Anforderungen von besonderer Bedeutung. Diese lässt sich mit Programmen wie z. B. Doors von Telelogics oder auch SysML-basierten Programmen realisieren (vgl. Kapitel 4.3).

Die gesamte Entwicklung basiert auf den in den Dokumenten enthaltenen lösungsfreien Anforderungen. Ihre Umsetzung wird in den jeweiligen Prozessphasen entwickelt und dokumentiert. Im Hinblick auf den durchgängigen Einsatz von Simulation bedeutet dies, dass für die Analyse von Systemen, Subsystemen und Komponenten Simulationsmodelle auf Basis der Entwicklungsdokumente erstellt werden. Die Simulationen dienen zur virtuellen Überprüfung der funktionalen Anforderungen. Damit kann festgestellt werden, ob das System bzw. Subsystem oder die Komponente den Anforderungen entsprechend entwickelt wurde. Dieses Vorgehen wird als Verifikation bezeichnet. Der Simulationseinsatz ermöglicht damit zusätzlich die frühzeitige virtuelle Verifikation im linken Strang des V-Modells (vgl. Abbildung 4-14). Die für die Erstellung der Simulationsmodelle benötigten Dokumente werden im Folgenden beschrieben.

Dokumentation der Entwicklung

In diesen Dokumenten wird die konkrete Umsetzung in den jeweiligen Entwicklungsphasen beschrieben. Im Gegensatz zu den Anforderungsdokumenten sind sie nicht lösungsfrei. Sie enthalten die Beschreibungen des Systemkonzeptes, der Subsysteme und Komponenten. In den Entwicklungsdokumenten wird konkret dargestellt, wie die Anforderungen realisiert werden. In Abbildung 4-14 sind die Entwicklungsdokumente der verschiedenen Entwicklungsphasen dargestellt.

Die Dokumentation der Systemarchitektur beschreibt das für die Weiterentwicklung ausgewählte Konzept. Ausgehend von den Hauptsystemanforderungen (Abk. HSAD) erfolgt in der Konzeptphase die Erstellung der Systemanforderungen (Abk. SAD). Während der Systemdetaillierung werden weitere Entwicklungsdokumente, in denen die Umsetzung der Subsysteme beschrieben wird, erstellt. Diese Dokumente sowie die Systemanforderungen dienen wiederum zur Erstellung der technischen Spezifikationen (Abk. TSD). Von den Zulieferern bzw. nach der Komponentenentwicklung werden Dokumente zu den einzelnen Komponenten erstellt. Anschließend erfolgt die Dokumentation der Tests.

Das Diagramm zeigt das V-Modell der Softwareentwicklung und -testung. Die linke Seite des 'V' stellt die Entwicklungsphasen dar: Konzeption (mit M4), Detaillierung (mit M5, M6, M7) und Komponenteneentwicklung. Die rechte Seite stellt die Testphasen dar: Flugtests (M11), Bodentests (M10), Integrationstests (M9), Funktionstests und Gerät-, Systemtests (M8). Die horizontale Achse ist in drei Dokumentationsbereiche unterteilt: 'Dokumentation der Architektur' (links), 'Dokumentation der Tests' (rechts) und 'Dokumentation der Komponenten' (unten). Die vertikale Achse ist in drei Ebenen unterteilt: 'HSAD' (oben), 'SAD' (Mitte) und 'TSD' (unten). Die Symmetrie wird durch diagonale Pfeile 'Virtuelle Verifikation' und horizontale Pfeile 'Verifikation' verdeutlicht.

Aus den Anforderungsdokumenten sowie den Entwicklungsdokumenten werden somit die Anforderungen für die anschließende Entwicklungsphase aufgestellt. Zusätzlich dienen die Dokumentationen der Systemarchitekturen zur Erstellung von Simulationsmodellen. Die Modelldetaillierung entspricht der Architekturebene (vgl. Abbildung 2-7). Die Dokumentationen der Subsysteme werden ebenfalls für die Erstellung von Modellen der Verhaltens bzw. Funktionsebene verwendet usw. Mithilfe dieser Modelle ist die virtuelle Verifikation der funktionalen Haupt- und Systemanforderungen sowie der Spezifikationen möglich.

Für die einfache Wiederverwendung von Modellen sowie den reibungslosen Einsatz in systemübergreifenden Simulationen ist zusätzlich zu den Modellen eine Modelldokumentation in der Informationsbasis zu hinterlegen. Folgende Punkte müssen in der Modelldokumentation enthalten sein:

- Zielsetzung der Simulation (Verwendungszweck)
- Modellzugehörigkeit
- Simulationsprogramm (Version, Hersteller usw.)
- Modellelemente und -struktur
- Modellvariablen
- Modellein- und -ausgänge
- Modellparameter
- Allgemeine Modellbeschreibung (mathematische Grundlagen, Annahmen usw.)
- Status des Modells (z. B. fehlerhaft, Version, Datum usw.)
- Validierungsstatus (z. B. Abgleich mit Messungen)
- Kontaktdaten (Bearbeiter/Ansprechpartner/Ersteller)
- Simulationsumgebung und -parameter
- Schnittstellendokumentation
- Abstraktionslevel des Modells
- Arbeitsbereich der Variablen und Parameter
- Verweis auf Simulationsergebnisse, Modell, Schnittstellendokumentation
- Zugehöriger Entwicklungsbereich (z. B. ATA-Kapitel)

66

Versionsstände und Änderungen. Zudem ergibt sich durch die Versions-Kommentare eine lückenlose Liste von Änderungen für die Dokumentation der Datei. Die Versionisierung muss ebenfalls für die Ablage der Modelle eingesetzt werden. Sie führt zu einer verbesserten Nachvollziehbarkeit und erhöht die Qualität der Modelle [134]. Für die Verwaltung und konsistente Speicherung der unterschiedlichen Versionen von Dokumenten, Programmen usw. werden in der Praxis Versionsmanagementsysteme eingesetzt. In der Informationsbasis kontrolliert das Versionsmanagement die Bearbeitung von Dokumenten. Wird ein Dokument bearbeitet, erlaubt das Versionsmanagement dritten nur den Lesezugriff auf die Dateien. Nach der Bearbeitung wird das Dokument mit einer neuen Versionsnummer versehen und wieder für alle freigegeben. Hierdurch bleiben Dokumente konsistent, und die Bearbeitungsschritte sind nachvollziehbar. Die Technologie des Versionsmanagements ist hinreichend bekannt, ihr Einsatz wird in diesem Konzept vorausgesetzt. Für die Programmierung des Datenaustauschsystems in Kapitel 5.3.1 wurde ebenfalls die CVS-Software eingesetzt. Für weitere Informationen zum Thema Versionsmanagement sei an dieser Stelle auf folgende Quelle verwiesen [134].

Zusätzlich zu den Modelldokumentationen werden Ergebnisdaten bereits durchgeführter Simulationen in der Informationsbasis abgelegt. Diese Daten können als Parameter für weitere Simulationsstudien sowie in losen Simulationskopplungen eingesetzt werden.

Für die systemübergreifenden Simulationen sind die Schnittstellen der Modelle von besonderem Interesse. Wie im Kapitel 4.1.2.2 beschrieben, werden für den Austausch und die Integration der Modelle zusätzliche Schnittstellendokumente benötigt, auf die in der Modelldokumentation ebenfalls verwiesen werden muss. In der IB werden von den an der Entwicklung Beteiligten beim Einstellen von Informationen (z. B. Dokumente, Modelle, Simulationsergebnisse usw.) Metainformationen hinterlegt. Diese Informationen erleichtern die Suche in der IB. Dokumentationen sind erforderlich, um Datentransparenz zu gewährleisten. Für jeden an der Entwicklung Beteiligten muss anhand der Dokumentation ersichtlich sein, welche Daten von wem und wann verwendet wurden. Das erfordert die konsequente Dokumentation aller notwendigen Informationen sowie die Bereitstellung der erstellten Dokumentationen in einem für alle Beteiligten zugänglichen Informationsraum.

4.2.2.2 Weiterverwendung von Modellen und Daten

Im derzeitigen Entwicklungsprozess entfallen ca. 50 % der Modellerstellung auf die Datenbeschaffung [108]. Dieser Anteil ist im Vergleich zur eigentlichen wertschöpfenden Tätigkeit sehr hoch. Zusätzlich erschwert die redundante Datenhaltung die Nachvollziehbarkeit. Oftmals ist es nur mit erheblichem Zeitaufwand möglich herauszufinden, mit welchem Datenstand (Variante, Version) die Berechnungsergebnisse erzielt wurden. Aus diesem Grund ist neben der Bereitstellung von Informationen in Form von Dokumenten der Austausch von Modellen und Daten zu fördern. Ein intensivierter Datenaustausch führt zu einer Reduzierung der Missverständnisse und somit zu einer erhöhten Konsistenz im Entwicklungsprozess. Die Wiederverwendung von Modellen verringert zusätzlich die Mehrfacherstellung von Modellen. Die nötigen Rahmenbedingungen für die Weiterverwendung von Modellen und Daten sollen im Folgenden vorgestellt werden.

Datenweiterverwendung

In der Entwicklung entstehen kontinuierlich Daten, die schnellstmöglich kommuniziert werden müssen. Durch die Bereitstellung der aktuellen Daten wird es möglich, die von diesen Daten beeinflussten Teilsysteme zeitnah anzupassen, ohne dass große Änderungsschleifen erforderlich werden. Es ist darauf zu achten, dass nur die Mitarbeiter, die ein Interesse an diesen Daten haben, über diese Aktualisierung informiert werden. Individuelle Zugriffsrechte dienen zur automatischen Generierung von Benachrichtigungen. Werden aktuelle Informationen in der IB eingestellt, dann erhalten die Mitarbeiter mit Zugriffsrecht auf diese Informationen eine Benachrichtigung. Es werden also immer die aktuellen Informationen kommuniziert und gleichzeitig wird eine Datenüberflutung der Mitarbeiter verhindert. Für die automatische Benachrichtigung existieren bereits viele kommerzielle Programme. So können z. B. mit dem Programm eRoom des Unternehmens Documentum (vgl. Kapitel 7.4.1) vom Administrator Zugriffsrechte auf die im eRoom enthaltenen Dateien und Ordner vergeben werden. Der Nutzer dieser Software kann automatische Benachrichtigungen für die ihm zugänglichen Ordner eigenständig konfigurieren [46]. Die Vergabe der Zugriffsrechte sowie alle weiteren administrativen Tätigkeiten für die Datenbereitstellung werden von der Koordinationsfunktion

4.2 Informationsmanagement

übernommen. In Kapitel 5.3.1 wird ein Algorithmus für die automatische Rechtevergabe vorgestellt.

Zusätzlich zu den Benachrichtigungen können Daten vom Mitarbeiter in der Informationsbasis auch aktiv gesucht werden. Diese Suche basiert auf den Metainformationen, die beim Ablegen der Daten in der IB hinterlegt werden müssen. Folgende Punkte müssen in den Metainformationen für Ergebnisdaten von Simulationen enthalten sein:

- Name der Daten bzw. Datei
- Modellzugehörigkeit
- Erstellungsdatum
- Verweis auf die Modelldokumentation und Dokumentation der Simulationsstudie
- Verknüpfung zum Simulationsmodell
- Zugehöriger Entwicklungsbereich (z. B. ATA-Kapitel)
- Kontaktdaten (Bearbeiter/Ansprechpartner/Autor)
- Kurze Ergebnisdatenbeschreibung
- Status

Zu den Ergebnisdaten müssen Dokumente der Simulationsstudie erstellt und in der IB abgelegt werden. Die Dokumentenstruktur ist analog zur Modelldokumentation (vgl. Kapitel 4.2.2.1). Die in den Metainformationen enthaltenen Verweise auf die Modelldokumentation, die Dokumentation der Simulationsstudie und das Simulationsmodell stellen die nötigen Hintergrundinformationen für das Verständnis der Ergebnisdaten bereit. Angaben über den Stand der verfügbaren Daten ermöglichen dem Nutzer zu entscheiden, wie verlässlich die Daten sind, sodass die Qualität der Berechnungen und Ergebnisse eingeschätzt werden kann.

Berechnungsdaten vorangegangener Simulationsstudien eignen sich für die lose Simulationskopplung und Parameterbereitstellungen in Simulationen. Sie tragen zur Reduzierung von Annahmen und Schätzwerten bei und erhöhen damit die Qualität anschließender Simulationsstudien.

Modellweiterverwendung

Einen wesentlichen Anteil für den Erfolg eines Konzeptes für die durchgängige Simulationsunterstützung in der Flugzeugsystementwicklung stellt die Weiterverwendung von Modellen dar. Im Hinblick auf die Weiterverwendung von Modellen kann unterschieden werden zwischen:

Modellweiterverwendung	Ein bestehendes Modell wird an neue Anforderungen angepasst.
Modellwiederverwendung	Ein bestehendes Modell wird erneut in einer Simulationsstudie verwendet.

Die erstellten Modelle sind mit den dazugehörigen Dokumentationen in der Informationsbasis abzulegen. Analog zum Einstellen der Dokumentationen und Daten müssen für die Modelle ebenfalls Metadaten in der IB hinterlegt werden. Die Informationen sind ähnlich aufgebaut wie bei den Daten, allerdings wird auf die Modelldokumentation, die Dokumentation der Simulationsstudie und die Simulationsdaten verwiesen. Ein Beispiel für Metainformationen zu einem Modell ist in Abbildung 5-33 (Entity-Relation) dargestellt.

Ein weiterer Unterschied besteht darin, dass die Modellabstraktionsebene hinterlegt werden muss. Die Abstraktionsebenen korrespondieren mit den Entwicklungsphasen, d. h., in der Konzeptphase werden Architekturmodelle, in der Detaillierungsphase Verhaltens- sowie funktionale Modelle und in der Komponentenentwicklung Komponentenmodelle erstellt. Der Modellabstraktionsgrad muss ebenfalls im Modellnamen auftauchen, ein Beispiel hierfür ist:

`Modellname_Abstraktionsebene_Version.*`

Diese Klassifizierung von Modellen gibt nicht nur Aufschluss über den Entwicklungsstand, vielmehr sind die Genauigkeit sowie die Komplexität der Modelle direkt erkennbar. Die Modelle der Architekturebene haben den Vorteil, dass sie geringe Anforderungen an die Implementierung sowie an die Hardwareplattform stellen. Durch ihren Einsatz kann z. B. die

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

Simulationszeit stark reduziert werden. Allerdings sind diese Modelle ungenauer als Modelle mit höherem Detaillierungsgrad.

Zusätzlich zur Modelldokumentation müssen im Modell Kontaktdaten, Erstellungsdatum, Version, Modellabstraktionsebene usw. enthalten sein. Für Modelle, die in Form von Modellcode vorliegen, sind diese Informationen direkt im Code zu integrieren (siehe Abbildung 4-8). Im Fall von Modellen, die mit einer spezifischen Modellierungssprache in grafischer Form erstellt wurden, sind diese Informationen ebenfalls zu hinterlegen. In Matlab/Simulink ist es z. B. möglich, diese Informationen in einen Textblock (Doc-Block) im grafischen Modell zu integrieren.

Basierend auf der Analyse des Entwicklungsprozesses in Kapitel 3 wird im Folgenden die Modellweiterverwendung bzw. -wiederverwendung dargestellt. Durch die Bereitstellung der Modelle in der Informationsbasis ist es möglich, existierende Modelle kontinuierlich in allen Phasen der Entwicklung weiterzuverwenden. Weitere Punkte für die Modellwieder- bzw. -weiterverwendung sind in Abbildung 4-15 dargestellt.

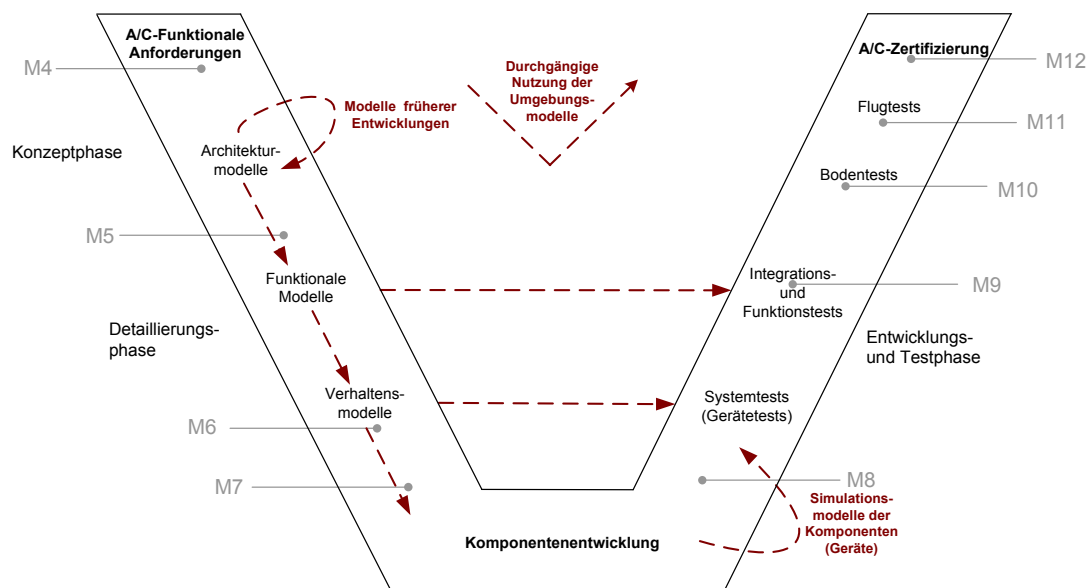


Abbildung 4-15 Weiter- bzw. Wiederverwendung von Simulationsmodellen

Sämtliche im Entwicklungsprozess erstellten Modelle stehen zur Wieder- bzw. Weiterverwendung in nachfolgenden Systementwicklungen zur Verfügung. Hierfür ist es lediglich erforderlich, dass die Modelle in der IB hinterlegt werden. Zudem ist bei der Wiederverwendung der Modelle darauf zu achten, dass Modelle dem Entwicklungsstand entsprechend eingesetzt werden. In der Konzeptphase sollten Architekturmodelle aus früheren Entwicklungen eingesetzt werden, und in der Detaillierungsphase führt der Einsatz von bestehenden funktionalen bzw. von Verhaltensmodellen zu einer Reduzierung des Modellierungsaufwands. Gleiches gilt für die Komponentenmodelle, die in der späten Detaillierungsphase nachfolgender Entwicklungsprozesse eingesetzt werden. Zusätzlich eignen sich die Modelle der Architektur-, Funktions- und Verhaltensebene für den Einsatz in späteren Entwicklungsphasen. Dies ist z. B. dann der Fall, wenn das zu untersuchende System von einem anderen System im geringen Maße beeinflusst wird. Hier ist es aus Gründen der Simulationszeitreduzierung sinnvoll, das System mit geringem Einfluss auf einem höheren Abstraktionsgrad zu simulieren. Damit ermöglichen die Modellabstraktionsebenen eine einfache Anpassung von Simulationsstudien an individuelle Anforderungen.

Auf Basis der Verhaltensmodelle wird die technische Geräte- bzw. Komponentenspezifikation erstellt. Die Komponenteneentwicklung erfolgt zum Großteil bei Zuliefererunternehmen. Aus diesem Grund ist bei der Bereitstellung von Entwicklungsdaten (z. B. Modelle und Informationen) für externe Unternehmen auf die Übertragungssicherheit zu achten (siehe Kapitel 4.2.2.3). Zudem muss der Zugriff externer Unternehmen auf die Informationsbasis wesentlich stärker reglementiert werden, um dem Wissensverlust des Flugzeugherstellers vorzubeugen.

Des Weiteren können sämtliche Modelle in der Testphase wiederverwendet werden. Für die Einbindung der Modelle in die HIL-Simulationen ist es allerdings erforderlich, dass diese echtzeitfähig sind. Die Komponentenmodelle und Verhaltensmodelle werden in den

4.2 Informationsmanagement

Systemtests eingesetzt. Zusätzlich dienen die Verhaltensmodelle der Fachabteilungen zum frühzeitigen Aufbau und zur Verifikation der Testvorrichtung. In der Phase der Integrationstests werden zur Überprüfung der Systeminteraktionen funktionale Modelle eingesetzt. Die Genauigkeit dieser Modelle ist für diese Phase ausreichend, da die Einzelsysteme bereits in den Systemtests verifiziert wurden. Um die Echtzeitfähigkeit der Modelle zu gewährleisten, kann es sinnvoll sein, Architekturmodelle oder tabellenbasierten Modelle in den Testphasen einzusetzen, um so die benötigte Rechenleistung zu reduzieren.

Einen Sonderfall stellt das Umgebungsmodell dar, das die Randbedingungen an den betrachteten Systemgrenzen abbildet. Eine Standardisierung der Umgebungsbedingungen ist erforderlich, damit jeder Entwickler auf Basis der gleichen Randbedingungen das entsprechende System entwickelt und es somit nicht zu Missverständnissen aufgrund von unterschiedlichen Annahmen des Umgebungsverhaltens kommt. Es ist sinnvoll, zum einen Umgebungsmodelle für die alleinstehenden, auf ein System begrenzten Simulationen zu verwenden und zum anderen Umgebungsmodelle für die systemübergreifenden Simulationen zu erstellen. Die Umgebungsmodelle sind parallel zur Entwicklung von der Koordinationsfunktion zu erstellen, fortlaufend zu detaillieren und für alle beteiligten Entwickler zugänglich in der Informationsbasis abzulegen, um so kontinuierlich in allen Entwicklungsphasen weiterverwendet werden zu können.

Die Voraussetzungen für die Daten- und Modellweiterverwendung bilden die organisierte Ablage der Informationen sowie der Zugang aller an der Entwicklung beteiligten Entwickler zu der entsprechenden (Daten-)Informationsbasis. Damit die Dokumentationen und Modelle von anderen Beteiligten gelesen und genutzt werden können, muss das Format der Dokumentationen von anderen lesbar und damit austauschbar sein. Modelle müssen mit einer Schnittstelle, zum Beispiel in C-Code (vgl. Abschnitt 2.2.3), bereitgestellt werden. Zusammenfassend ergeben sich folgende Voraussetzungen für den Modellaustausch:

- Dokumentation der Entwicklung, der Daten und der Modelle
- Standardisierte Datenformate (z. B. PDF-Dateien)
- Standardisierte Austauschformate (z. B. XML, HTML usw.)
- Modellschnittstellen
- Bereitstellung der Informationen
- Abteilungsübergreifende Kommunikation.
- Versionsmanagement

Die Gestaltung der Ablage und des Zugriffs auf die Informationsbasis wird im folgenden Kapitel dargestellt.

4.2.2.3 Organisation der abteilungsübergreifenden Informations-bereitstellung

Das Konzept der Informationsbereitstellung wird sich nur dann durchsetzen, wenn bestehende Infrastrukturen weiterverwendet werden können. Zusätzlich ist der Erfolg der Informationsbasis von der Akzeptanz der Nutzer abhängig. Aus diesem Grund sind nutzerspezifische Anforderungen, wie die einfache Informationsbeschaffung, eine zentrale Informationsbereitstellung, die Erweiterbarkeit sowie der Zugriff auf alle relevanten Informationen bei gleichzeitiger Vermeidung der Informationsüberflutung, zu realisieren. Diese Anforderungen lassen sich wie folgt einteilen:

1. Nutzung bestehender Infrastrukturen
2. Informationszugriff
3. Struktur der Informationsbereitstellung
4. Abrufen und Einstellen von Informationen
5. Datensicherheit

1. Nutzung bestehender Infrastrukturen

Das Konzept der Informationsbasis basiert auf der Weiterverwendung bestehender Infrastrukturen für die Informationsbereitstellung im Unternehmen. Daher ist es erforderlich, dass eine Möglichkeit existiert, verschiedene Informationsquellen in der IB zu integrieren. Für jede Informationsquelle (Netzlaufwerke, Dateisysteme, Datenbank, PDM-Systeme usw.) wird ein Server benötigt. Um dem Benutzer die Auswahl des richtigen Servers zu erleichtern, wird ein

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

Master-Server eingesetzt, der je nach Anfrage dem Client mitteilt, wo sich die angeforderten Daten befinden (siehe Abbildung 4-16).

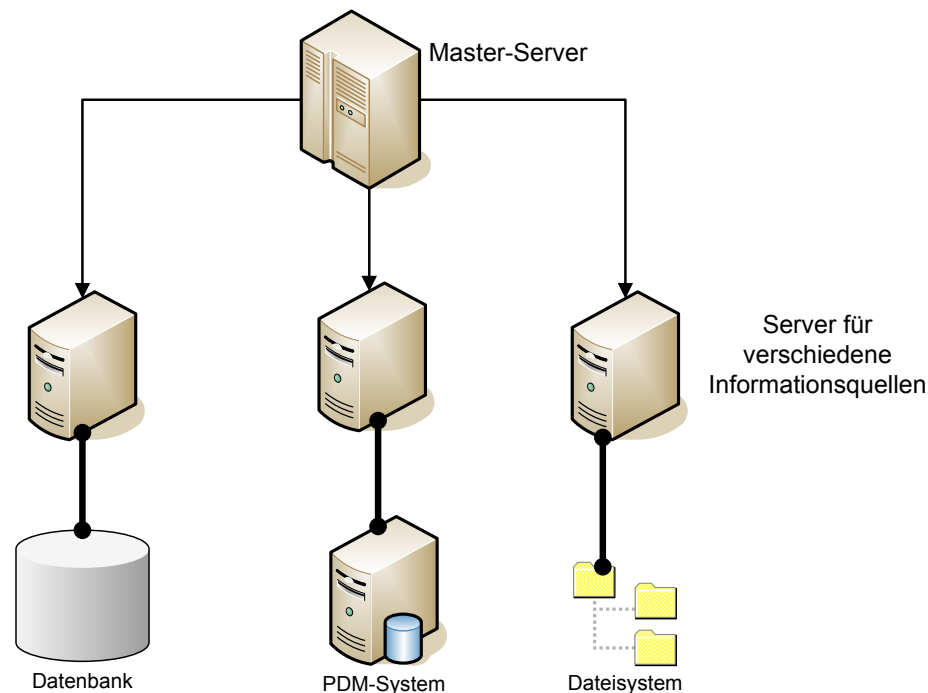


Abbildung 4-16 Server-Topologie

Dafür gibt der Master-Server eine Referenzliste mit allen infrage kommenden Servern an den Client zurück. Der Master-Server dient dazu, dem Client mitzuteilen, an welche Server dieser die Suchanfrage stellen muss. Der Client stellt anhand der Referenzlisten eine Suchanfrage an jeden einzelnen Server und bereitet die Ergebnisse für den Benutzer auf (siehe Abbildung 4-17).

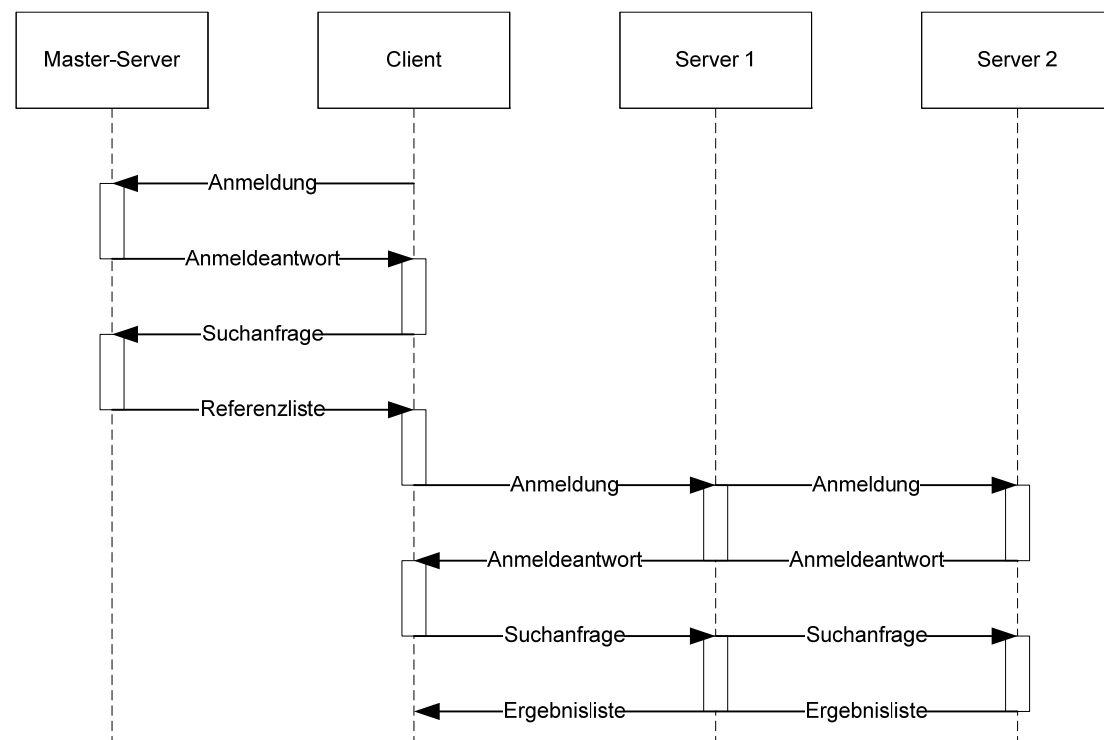


Abbildung 4-17 Sequenzdiagramm für die Informationssuche

Die Suche erfolgt auf den Servern dezentral, sodass der Master-Server entlastet wird. Für die Suche ist es notwendig, dass Metainformationen zu den Daten, Dokumenten und Simulati-

4.2 Informationsmanagement

onsmodellen in der jeweiligen Informationsquelle hinterlegt werden. Die Metainformationen sind direkt mit den Daten verbunden, wodurch eine konsistente Datenverwaltung gewährleistet wird. Werden z. B. Daten aus der Informationsquelle entfernt, werden auch die dazugehörigen Metainformationen entfernt, sodass diese Daten in der Suche nicht mehr auftauchen. Würden die Daten von den Metainformationen getrennt, um z. B. eine zentrale Suche auf dem Master-Servers durchzuführen, bestünde das Problem der Datenkonsistenz. Für die Änderung einer Datei in einer Informationsquelle müsste zusätzlich die Metainformation des Master-Servers geändert werden. Das Löschen einer Datei könnte dazu führen, dass der Verweis auf die Quelle gefunden wird, diese aber nicht mehr existiert (bei HTTP z. B. Fehlercode 404). Nachdem die Suchanfrage auf den Servern dezentral durchgeführt wurde, wird eine Ergebnisliste mit Verknüpfungen zu den gesuchten Informationen an den Client gesendet und dem Nutzer zur Verfügung gestellt. Wählt der Nutzer eine Datei aus der Ergebnisliste aus, wird diese direkt über eine Verknüpfung von der jeweiligen Informationsquelle heruntergeladen. Dieses Vorgehen für die Bereitstellung von Informationen wird bereits von Internetsuchmaschinen wie dem MetaCrawler genutzt [110]. Einen Vorteil dieses Vorgehens stellt neben der Lastverteilung bedingt durch die dezentrale Suche, die einfache Erweiterbarkeit des Systems dar. Wird ein neuer Server in der Entwicklung eingeführt oder ein alter Server abgeschaltet, muss lediglich die Server-Referenzliste des Master-Servers angepasst werden.

Theoretisch ist es auch möglich, dass der Master-Server die Daten für den Client beschafft und zur Verfügung stellt. Allerdings entsteht dabei eine doppelte Netzwerkbelastung, da die Daten vom Server an den Master-Server und vom Master-Server an den Client anstatt direkt vom Server an den Client gesendet werden.

Für die Weiterleitung auf andere Server eignet sich das HTTP-Protokoll. Dieses Protokoll kann dafür verwendet werden, einen Simulationsdatenserver (Master-Server) aufzusetzen, der keine Informationen bereitstellt, dafür aber die Weiterleitungen zu anderen Servern anbietet, die dasselbe Protokoll implementieren. In Kapitel 5.3 wird die Netzwerkkommunikation einer losen Simulationskopplung mithilfe eines HTTP-Protokolls realisiert. Zusammenfassend lässt sich sagen, dass durch den Einsatz eines Master-Servers eine zentrale Informationsbereitstellung unter Einbeziehung der bestehenden Infrastruktur realisiert werden kann.

2. Informationszugriff

Ein Großteil der Koordinationstätigkeiten entfällt auf den Bereich des Informationsmanagements. Zu diesen administrativen Aufgaben gehören die Überprüfung der in der IB hinterlegten Daten, die Speicherverwaltung und die Vergabe von Zugriffsrechten. Bedingt durch die zentrale Informationsbereitstellung vereinfacht sich der Aufwand für die Rechtevergabe erheblich, da diese nur im zentralen Knoten, dem Master-Server, hinterlegt werden müssen. Über die Zugriffsrechte werden die für den Mitarbeiter relevanten Informationen bereitgestellt. Für die Verwaltung dieser Rechte wird für jeden Nutzer der IB ein Datenbank-eintrag angelegt. In diesem Eintrag sind der Name, Kontaktinformationen, ein eindeutiger Identifikator und ein Passwort hinterlegt (siehe Kapitel 5.3.1). Der Nutzer meldet sich einmal zentral an der IB an, dies kann z. B. mit der Verzeichnisdienstsoftware Novell von eDirectory oder Active Directory Account von Microsoft erfolgen [120]. In diesem Fall hat der Nutzer zwar keinen Gesamtüberblick über alle vorliegenden Informationen, sieht allerdings alle für ihn interessanten Informationen, ohne von einer Fülle an Informationen überflutet zu werden. In den ersten zwei Punkten der abteilungsübergreifenden Informationsbereitstellung wurde der zentrale Zugriff auf individuelle Informationen beschrieben. Im folgenden Abschnitt wird beschrieben, wie die Informationen in der IB strukturiert werden.

3. Struktur der Informationsbereitstellung

Um den Nutzer bei der Beschaffung und beim Ablegen von Informationen zu unterstützen und die Rechtevergabe zu vereinfachen, ist es sinnvoll, Informationen in der IB strukturiert abzulegen. In Abbildung 4-18 ist die Struktur der Informationen dargestellt.

Die gesamte Informationsmenge wird zum einem in die einzelnen Entwicklungsbereiche (ATA-Kapitel) unterteilt und zum anderen zeitlich in die einzelnen Entwicklungsphasen (Konzeption, Detaillierung, Komponentenentwicklung und Tests). Die Detaillierung der zur Verfügung gestellten Informationen nimmt mit voranschreitender Entwicklung bis zur Testphase zu. Die Unterteilung in die Phasen wird als Sicht 1 bis 4 bezeichnet, beispielsweise steht die Sicht 1 für die Konzeptphase.

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

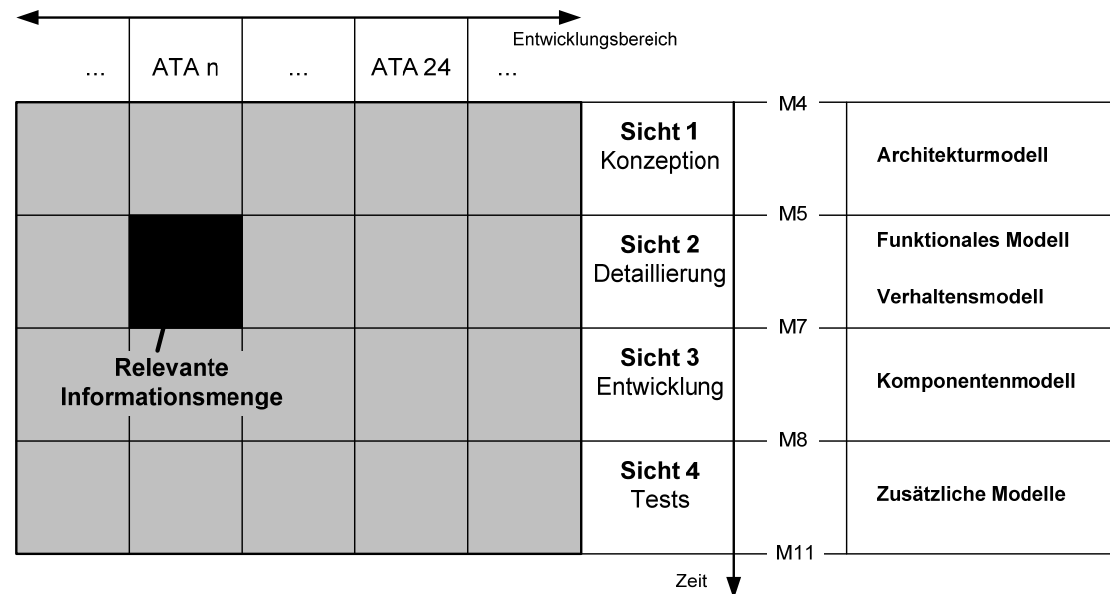


Abbildung 4-18 Strukturierung der Informationen

Die Sicht beinhaltet Informationen (Dokumente, Modelle, Daten usw.) der entsprechenden Entwicklungsphase. Ein Dokument wird z. B. in dem dazugehörigen Entwicklungsbereich und der dem Entwicklungsprozess entsprechenden Sicht hinterlegt. Das Dokument steht damit allen Mitarbeitern, die Zugriffsrechte für diese Sicht des Entwicklungsbereichs in der IB besitzen, zur Verfügung.

Die Simulationsmodelle korrespondieren dabei ebenfalls mit den Entwicklungsphasen. In der Sicht der Konzeptphase werden Architekturmodelle, in der Detaillierungsphase werden funktionale und Verhaltensmodelle, in der Komponentenentwicklung werden Komponentenmodelle und in der Testphase werden zusätzliche Modelle verschiedener Detaillierung hinterlegt (vgl. Abbildung 2-7). Für das Aufrufen des Lese- bzw. Schreibbereiches sind die in Abbildung 2-9 dargestellten vier Schritte nötig.

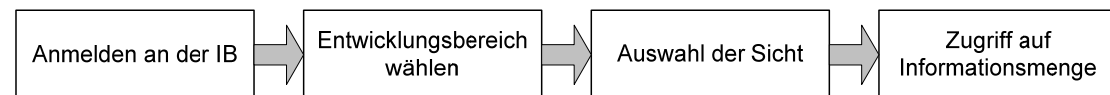


Abbildung 4-19 Vorgehen bei der Auswahl einer Informationsmenge

Nach dem Anmelden an der Informationsbasis wählt der Nutzer zunächst den entsprechenden Entwicklungsbereich (z. B. A/C-Programm, ATA-Kapitel, Sub-ATA-Kapitel) und anschließend die benötigte Sicht aus. Durch dieses Vorgehen findet die oben beschriebene Selektion der Informationen statt, und der Nutzer gelangt in den individuellen Informationsbereich, zudem unterstützt eine Suchfunktion die Navigation in der Informationsbasis. In Kapitel 5.3.1 wird die Umsetzung der in Abbildung 4-19 dargestellten Struktur in Form eines Baumdiagramms realisiert (vgl. Abbildung 5-37).

4. Abrufen und Einstellen von Informationen

Im vorangegangenen Kapitel wurde beschrieben, wie der Nutzer in der Informationsbasis navigieren kann. Wurden neue Informationen erstellt, so müssen diese im jeweiligen Entwicklungsbereich und der dem Entwicklungsprozess entsprechenden Sicht eingestellt werden. Für das Einstellen werden Schreibrechte benötigt, damit ist das Einstellen von Informationen direkt von den individuellen Zugriffsrechten abhängig. Zusätzlich zu den eigentlichen Informationen müssen Metainformationen hinterlegt werden. Diese erleichtern zum einen den Umgang mit den Informationen und unterstützen zum anderen die Suchfunktion der IB. Die eingestellten Modelle, Dokumente und Daten müssen den in Kapitel 4.2 beschriebenen Richtlinien entsprechen. Die Dokumente müssen standardisiert sein, um von allen Nutzern verwendet werden zu können (z. B. PDF- oder HTML-Format). Die Modelle für die systemübergreifenden Simulationen sind mit den entsprechenden Schnittstellen bereitzustellen (vgl. Kapitel 4.1). Für Daten ist eine automatische Validierung beim Einstellen in die IB möglich. In Kapitel 5.3.1 wird aus diesem Grund die automatische Validierung von

4.2 Informationsmanagement

Simulationsdaten realisiert. In der Informationsbasis werden vom Nutzer somit vordefinierte Informationen in einem vordefinierten Bereich hinterlegt.

Durch die Vorgabe von festgelegten Zeitpunkten zur Ablage der Daten (z. B. Dokumente, Modelle, usw.) in die Informationsbasis stehen die Informationen für die Weiterverwendung in der Entwicklung zur Verfügung. Allerdings besteht die Gefahr bei einer Datenablage durch den Nutzer darin, dass die Daten verspätet oder gar nicht in der IB eingestellt werden. Aus diesem Grund kontrolliert die im Unternehmen implementierte Koordinationsfunktion, ob die für die jeweilige Entwicklungsphase erforderlichen Daten, Modelle und Dokumente von den Fachabteilungen vollständig und korrekt in der IB eingestellt wurden. Beim Einstellen neuer Informationen erhalten alle Mitarbeiter, die zum Lesezugriff auf die Information befugt sind (Schreibzugriff beinhaltet den Lesezugriff), eine Benachrichtigung. Diese Benachrichtigungen können wie bei eRoom (vgl. 7.4.1). individuell von Nutzer konfiguriert werden. Um das Überschreiben älterer Versionen sowie die Historie der in der IB enthaltenen Daten nachvollziehen zu können, werden Versionsmanagementsysteme eingesetzt. Mithilfe von PPS-Systemen werden Fertigungen bestimmter Versionen der Produktstruktur umgesetzt [173].

In Abbildung 4-19 ist dargestellt, wie der Nutzer zu den individuellen Informationen navigieren kann. Neben dieser Möglichkeit existieren zwei weitere Möglichkeiten zur Informationsbeschaffung. Zum einen können die Mitarbeiter die Verknüpfungen der automatischen Benachrichtigungen zum Abrufen aktueller Informationen nutzen, zum anderen kann die Suchfunktion der Informationsbasis eingesetzt werden. Die Akzeptanz bei den Mitarbeitern ist stark von der Effizienz der Informationsbeschaffung abhängig. In einer umfangreichen Informationsmenge ist die Suchfunktion von besonderer Bedeutung. Um die Suche in angemessener Zeit durchführen zu können, werden in der IB Metainformationen zu den Daten abgelegt.

Eine geeignete Suchfunktion muss die Metainformationen nicht nur nach einer Eigenschaft, sondern nach einer Kombination von Eigenschaften durchsuchen können, um so den Umfang der infrage kommenden Suchergebnisse übersichtlich zu gestalten. Die vom Nutzer an die Informationsbasis gestellten Anforderungen sind damit erfüllt. Allerdings muss die IB weiteren Anforderungen gerecht werden. Im folgenden Abschnitt wird auf die Gewährleistung der Datensicherheit eingegangen.

5. Datensicherheit

Der Austausch von Informationen über das Internet oder ein Unternehmensnetzwerk ermöglicht es unbefugten Netzteilnehmern, die Übertragung mitzuschneiden und die Daten daraus zu extrahieren. Werden die Daten verschlüsselt übertragen, ist es für Unbefugte kaum noch möglich, auf die Daten zuzugreifen. Für den designierten Empfänger, der im Besitz eines Schlüssels (z. B. Passwort) ist, ist weiterhin die uneingeschränkte Nutzung der Daten möglich. Der Schlüssel wird maximal einmal über das Netzwerk übertragen, wenn dem entsprechenden Benutzer der Schlüssel mitgeteilt wird. Um die Verschlüsselung der Übertragung zu implementieren, eignet sich z. B. das TLS-Protokoll (engl.: Transport Layer Protocol) [166]. Es baut einen verschlüsselten Tunnel auf, über den eine gewöhnliche HTTP-Verbindung in der Version 1.1 zustande kommen kann.

Um die IB vor automatisierten Passwortangriffen, wie z. B. dem Ausprobieren von Passwörtern, zu schützen, wird einem Client nur eine bestimmte Anzahl von gescheiterten Anmeldeversuchen pro Zeiteinheit gewährt. Dadurch wird die Anzahl der testbaren Passwörter pro Zeiteinheit deutlich reduziert (engl.: brutal force protection).

Häufig versuchen Angreifer, einen Server so stark zu belasten, dass Speicherüberlaufen oder Programmabstürze die Folge sind. Hierfür wird eine Vielzahl von Verbindungen zum Server aufgebaut. Da der Server gezwungen ist, für jeden verbundenen Client einen Thread zu starten, kann dieses zur Überlastung der Netzwerkverbindung führen. Verhindern lässt sich dies durch eine Limitierung der zugelassenen Verbindungen. Diese Limitierung ist vom Administrator, in diesem Konzept der Koordinationsfunktion, zu konfigurieren, da die kritische Threadanzahl von System zu System unterschiedlich ist.

4.2.2.4 Zusammenfassung der Informationsbereitstellung

Eine geeignete Informationsbereitstellung erhöht die Produktivität in der Entwicklung. In der Automobilindustrie wurde nachgewiesen, dass sich die Entwicklungszeit durch den Einsatz von Simulationsdaten-Managementsystemen um bis zu 25 % reduzieren lässt [17]. In diesem Kapitel werden die wichtigsten Punkte in Bezug auf die Informationsbasis zusammengefasst

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

und die Anforderungen an das Konzept verifiziert. Aus diesem Grund wird zunächst überprüft, ob die im Kapitel 3 aufgestellten Anforderungen an die Simulation im Prozess erfüllt werden (vgl. Tabelle 3-3).

Häufige, späte und teure Änderungen aufgrund von ungenauen Daten und Missverständnissen in der Entwicklung und damit große Iterationsschleifen können durch eine Intensivierung des systemübergreifenden Informationsaustauschs vermieden werden. Eine einheitliche grafische Benutzeroberfläche stellt die Schnittstelle zwischen dem Nutzer und der Informationsbasis dar. Die Informationen werden zentral bereitgestellt, das hat den Vorteil, dass der Nutzer sich nur einmal authentifizieren muss, um auf alle verfügbaren Daten zugreifen zu können. Ein Master-Server ermöglicht eine dezentrale Verteilung der Entwicklungsdaten. Dieses Konzept erlaubt es, bestehende Infrastrukturen weiterzuverwenden sowie weitere Informationsquellen zu integrieren. Dadurch können für die Datenablage immer die am besten geeigneten Informationsquellen eingesetzt werden. Die individuellen Zugriffsrechte verhindern die Informationsüberflutung und unterstützen die intuitive Navigation in der IB. Über automatische Benachrichtigungen werden den Mitarbeitern die aktuellen Informationen bereitgestellt. Standards für die Modellschnittstelle, Dokumente und Daten ermöglichen die einfache Weiterverwendung. Durch den Einsatz einer Koordinationsfunktion (z. B. Koordinationsteam oder IT-Lösung) wird der Datenfluss kontrolliert und die Einhaltung der Richtlinien überwacht. Das Konzept der Informationsbasis unterstützt den kontinuierlichen und konsistenten Austausch von Informationen während des gesamten Entwicklungsprozesses. In Tabelle 4-5 ist eine Zusammenfassung zur Umsetzung der Anforderungen an das Informationsmanagement durch den Einsatz der IB dargestellt.

4.2 Informationsmanagement

Anforderung: Informationsmanagement		Umsetzung
Intuitive Informationsbeschaffung	Einheitliche Benutzerschnittstelle	Zentraler Informationszugriff mit einheitlicher grafischer Benutzeroberfläche (Abk. GUI)
	Einheitlicher zentraler Informationszugriff	Zentrale Informationsbeschaffung über den Master-Server
	Einfache Informationsbeschaffung	Automatische Benachrichtigungen, eine Suchfunktion und das eigenständige Navigieren unterstützen den Benutzer
	Verhinderung der Datenüberflutung	Zugriffsrechte regeln die individuelle Informationsbereitstellung
	Automatische Benachrichtigung über relevante Informationen	Benachrichtigungen sind konfigurierbar
	Übersichtliche Verwaltungsstruktur	Einteilung in Entwicklungsbereiche und Entwicklungsschichten
Intensivierung des Informationsaustausches	Standardisierte Dokumentation von Simulationsstudien und Modellen	Richtlinien zur Bereitstellung der Dokumente (vgl. Kapitel 4.2.2.1)
	Standardisierte Ablage von Simulationsergebnissen	Richtlinien und Standards werden von der Koordinationsfunktion definiert
	Kontinuierliche Kommunikation von Informationen	Regeln für das Einstellen von Dokumenten, Modellen und Daten in den Entwicklungsphasen
	Zeitnahe Informationsbereitstellung	Automatische Benachrichtigungen
	Statusangabe der Informationen	Name, Datum, Version, Status usw. werden als Metainformationen hinterlegt
	Direkter Datenaustausch zwischen Simulationsprogrammen	Simulationskopplung (vgl. Kapitel 2.3.2.)
	Unternehmensweite Bereitstellung der benötigten Informationen	Benutzerzugang über zentralen Master-Server
	Ausreichende Speicherkapazität	Koordinationsfunktion stellt Speicherkapazität zur Verfügung
	Kontrolle der Informationen	Koordinationsfunktion kontrolliert die Informationseinstellung in der IB
	Individuelle Rechtevergabe	Zugriffsrechte werden durch die Koordinationsfunktion vergeben
	Konfigurationsmanagement und Versionsmanagement	Für das Konfigurationsmanagement und Versionsmanagement werden Standardsoftwareprogramme eingesetzt
	Einfache Informationsablage	Der Nutzer kann Daten nur in Bereichen mit Schreibberechtigung hinterlegen
	Informationen vor dem Zugriff unbefugter Personen schützen	Verschlüsselung und Begrenzung der Anmeldeversuche

Tabelle 4-5 Umsetzung der Anforderungen an das Informationsmanagement

Ein Leitfaden für die Einführung eines EDM/PDM-Systems, wie es die Informationsbasis darstellt, wird in der VDI-Richtlinie 2219 gegeben.

4.3 Standardisierte Modellierung

Wie in vorangegangenen Kapiteln beschrieben erweist sich die Simulation als hocheffizientes Kommunikationsmittel. Sämtliche Simulationen basieren auf Modellen, die ein Abbild der Wirklichkeit auf einem bestimmten Abstraktionsgrad repräsentieren. In der Entwicklung multidisziplinärer Systeme sind Experten verschiedenster Fachrichtungen beteiligt, die unterschiedliche Werkzeuge, Methodiken und Begriffswelten nutzen (vgl. Defizit 3 in Kapitel 3). In diesem Kapitel werden die derzeitige Systemmodellierung und die daraus resultierenden

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

Probleme beschrieben. Für den eindeutigen, unmissverständlichen Informationsaustausch wird eine standardisierte Systembeschreibung bzw. Modellierungssprache benötigt, die es ermöglicht, sämtliche Fachdisziplinen allgemein darzustellen.

Um die programmspezifischen Stärken weiterhin nutzen zu können und eine Monopolbildung eines Simulationsanbieters zu unterbinden, ist es von entscheidender Bedeutung, für den jeweiligen Anwendungsfall das am besten geeignete Simulationsprogramm zu nutzen. Ein Nachteil dieses Vorgehens ist die Verwendung unterschiedlicher Modellierungssprachen der einzelnen Simulationsprogramme, sodass eine fachdisziplinübergreifende, eindeutige und für alle Beteiligten verständliche Darstellung der Einzelsysteme nicht garantiert werden kann. Die Simulationsergebnisse dienen zwar als Kommunikationsmittel zwischen den einzelnen Fachdisziplinen, ein Verständnis der Einzelsysteme ist jedoch behindert, da syntaktisch unterschiedliche Modellierungssprachen genutzt werden. Hierdurch bedingt, erschließt sich den Beteiligten das System in eng gesteckten Systemgrenzen, das Verständnis für das Gesamtsystem geht jedoch verloren. Für die Entwicklung qualitativ hochwertiger Produkte ist das ganzheitliche Denken der Mitarbeiter von entscheidender Bedeutung. Das Verhalten eines Gesamtsystems ist nicht zwangsläufig aus der Summe seiner Einzelteile abzuleiten, oftmals beeinflussen Interaktionen der Einzelteile das Gesamtsystem auf unvorhergesehene Art und Weise. Das Gesamtproblem muss fokussiert werden, sodass die Summe der Teillösungen die bestmögliche Gesamtlösung ergibt [186]. Aus diesem Grund ist es von besonderer Bedeutung, bereits während der Modellierung ein systemübergreifendes Verständnis aller Beteiligten zu unterstützen. Das derzeitige Vorgehen in der Entwicklung von Flugzeugsystemen weist aber gerade diese tayloristische Verteilung der Aufgaben und damit die Entwicklung in engen Systemgrenzen auf. Jeder modelliert und simuliert nur in der eigenen Fachdisziplin, Systeminteraktionen werden kaum oder erst sehr spät berücksichtigt (vgl. Kapitel 3.1). Es werden Insellösungen generiert, und durch die unterschiedlichen Modellierungssprachen ist es den Systemabteilungen kaum möglich, Systeminformationen in Form von Simulationsmodellen auszutauschen. Eine Möglichkeit zur Lösung dieses Problems ist das Erlernen sämtlicher für den Entwicklungsprozess relevanter Modellierungssprachen aller Beteiligten. Dieses Vorgehen scheidet allerdings aus, da der zusätzliche Schulungs- und Weiterbildungsaufwand in keinem Verhältnis zum Nutzen steht. Ein weitaus ressourcenschonenderes Vorgehen besteht in der Verwendung einer allgemeingültigen Modellierungssprache. Im Vergleich zum derzeitigen Vorgehen ist zunächst ein erhöhter Einsatz von Ressourcen nötig, um die Sprache zu erlernen. Dieser Aufwand wird durch das ganzheitliche Denken der Mitarbeiter und die damit verbundene Produktqualitätssteigerung innerhalb kürzester Zeit amortisiert, zudem werden Systeminteraktionen schon während der Modellierung berücksichtigt und fehlerhafte Schnittstellen frühzeitig entdeckt. Aus diesem Grund bildet eine konsistente Modellierung während des gesamten Entwicklungsprozesses einen Grundpfeiler für das Konzept zur durchgängigen Unterstützung des Entwicklungsprozesses von Flugzeugsystemen. Im folgenden Abschnitt wird eine standardisierte Modellierungssprache vorgestellt, welche die in Kapitel 3 (vgl. Tabelle 3-2) aufgestellten Anforderungen an eine einheitliche Systembeschreibung erfüllt.

In der Systemsimulation haben sich verschiedene Modellierungssprachen, wie z. B. Modelica für mechatronische Systeme, OpenMast für elektrische Systeme und VHDL-AMS als Quasistandard in der Automobilindustrie, etabliert. Bei den vorgestellten Sprachen handelt es sich um textuelle Sprachen mit fachdisziplinabhängigen Modellierungseigenschaften [32], [167]. Für eine allgemeingültige Modellierungssprache besteht jedoch die Notwendigkeit, bis zu einem gewissen Grad abstrakt und generisch zu sein, um möglichst für sämtliche Fachdisziplinen verwendet werden zu können. Fachspezifische Modellierungseigenschaften würden den Einsatz der Sprache wiederum einschränken. Zusätzlich muss es möglich sein, die Sprache an die jeweilige Problemstellung anzupassen. Für das intuitive Verständnis weist eine grafische Modellierungssprache zusätzliche Vorteile gegenüber den textuellen Sprachen auf.

Im Bereich der Softwareentwicklung hat sich die Unified Modeling Language (Abk. UML) als Modellierungssprache etabliert [153]. Trotz erheblicher Anstrengungen wurde bis vor Kurzem im Bereich des Systems Engineering keine einheitliche Modellierungssprache entwickelt. Das führte zu erheblichen Reibungsverlusten in interdisziplinären Projekten. Aus diesem Grund entschied das International Council on Systems Engineering (Abk. INCOSE), die UML zu einer Standardsprache des Systems Engineering weiterzuentwickeln. Die UML erfüllt zum Großteil die in Kapitel 3 aufgestellten Anforderungen. Sie ermöglicht die grafische Modellierung, ist verbreitet, generisch und damit an spezifische Bedürfnisse anpassbar; es

4.3 Standardisierte Modellierung

existiert eine Vielzahl an Modellierungswerkzeugen, Literatur und Schulungen. Allerdings muss die UML in einigen Bereichen an die spezifischen Anforderungen des Systems Engineering angepasst werden. Die auf der UML 2.0 basierende Sprache mit dem Namen System Modeling Language (Abk. SysML) wurde in der Version 1.0 im April 2006 als Standard von der Object Management Group (Abk. OMG) angenommen (siehe Abbildung 4-20) [186].

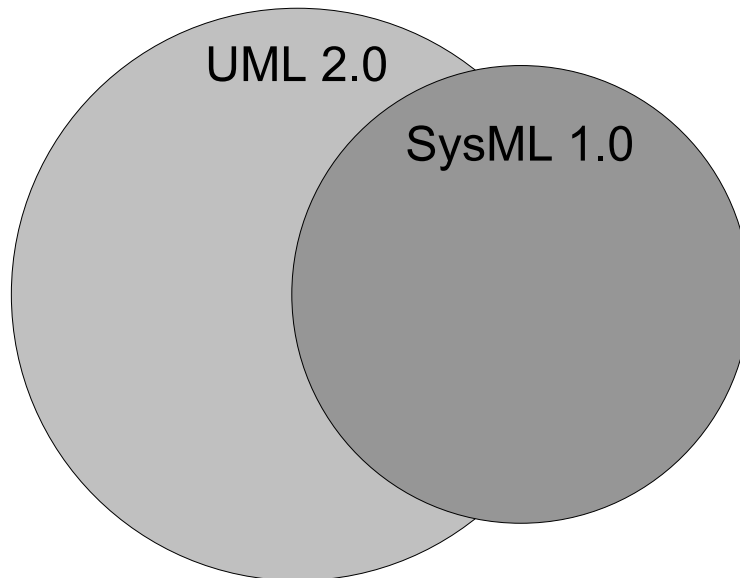


Abbildung 4-20 UML 2.0 als Basis der SysML [186]

Ihr Ziel ist es, dem Systemingenieur eine interdisziplinäre Modellierungssprache bereitzustellen, welche den gesamten Entwicklungsprozess von der Anforderungserstellung bis zum fertigen Produkt konsistent unterstützt und an die jeweiligen Bedürfnisse angepasst werden kann. Sie soll den Informationsaustausch der Beteiligten verbessern und damit Missverständnisse zwischen den verschiedenen Fachdisziplinen minimieren. Die Stärken von SysML liegen in der Aufnahme von Anforderungen sowie in der guten Verfolgbarkeit (engl.: traceability) von Aspekten, beispielsweise der Umsetzung von Anforderungen [186]. Es ist möglich, Modelle verschiedener Detaillierungsebenen zu erstellen, so können einfache Modelle mithilfe von Zusicherungsdiagrammen und komplexe Modelle mithilfe von Zustandsdiagrammen erstellt werden. In Abbildung 5-38 ist ein Zustandsautomat für die Zerlegung von Simulationsdaten und in Kapitel 7.6 sind ein Blockdiagramm eines Brennstoffzellensystems sowie ein Auszug aus den in Kapitel drei aufgestellten Anforderungen im SysML-Standard dargestellt. Der SysML-Standard ist bereits zum gegenwärtigen Zeitpunkt in vielen Softwareprogrammen integriert (z. B. Enterprise Architect, Telelogic Rhapsody, ARTiSAN, Embedded Plus, usw.) [165]. Allerdings werden ausschließlich zeitdiskrete und damit ereignisorientierte Simulationen in den Softwarewerkzeugen ermöglicht (vgl. Kapitel 2.1). Diese basieren auf Zustandsautomaten und können mithilfe von SysML-Zustandsdiagrammen modelliert und anschließend simuliert werden. Zeitkontinuierliche Simulationen werden zurzeit noch nicht von kommerziellen Anbietern unterstützt. Es wurden bereits erste Projekte erfolgreich beendet, die sich zum Ziel gesetzt haben, Modelle aus Modellierungssprachen, die für zeitkontinuierliche Simulation genutzt werden können, in SysML zu konvertieren und andersherum. Das Georgia Institute of Technology hat z. B. Modelica-Modelle der Fluidik-Bibliothek mithilfe eines sogenannten Wrappers (deutsch: die Hülle) in SysML-Modelle umgewandelt (siehe Abbildung 4-21). Die so erstellte SysML Fluid-Bibliothek kann wiederum zur Modellierung von SysML-Fluidik-Systemen eingesetzt werden, die dann ihrerseits automatisch in Modelica-Modelle umgewandelt werden können (siehe Abbildung 4-22) [125].

Zurzeit arbeiten verschiedenen Softwarehersteller an einer Einbindung von SysML in ihre Produkte. Für die Zukunft bedeutet dies, dass für die Modellierung multidisziplinärer Systeme nur noch eine Modellierungssprache benötigt wird, die in der jeweiligen Software in die programmspezifische Modellierungssprache konvertiert wird. Für den Entwicklungsprozess bedeutet die Nutzung einer universellen Modellierungssprache eine Optimierung der fachdisziplinübergreifenden Kommunikation sowie die Minimierung von Missverständnissen.

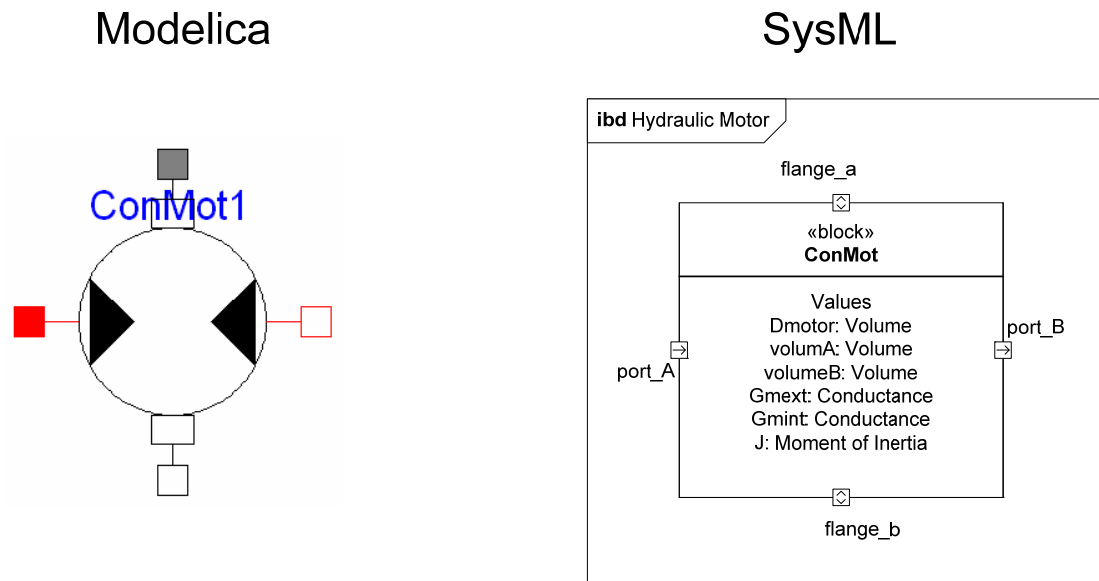


Abbildung 4-21 Umwandlung eines Modelica-Modells in einen SysML Block [125]

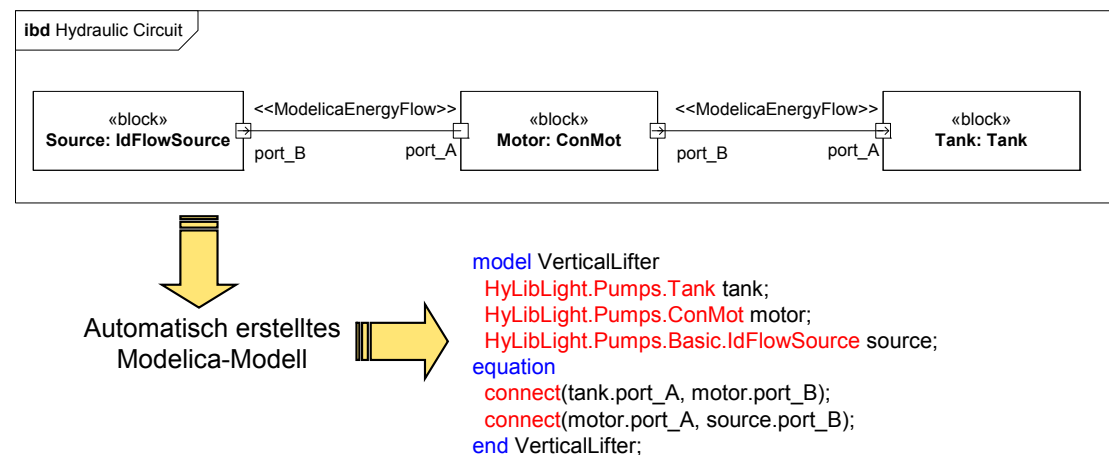


Abbildung 4-22 Automatische Umwandlung eines SysML-Modells in Modelica [125]

Zusätzlich trägt die konsistente Modellierung zum Gesamtsystemverständnis jedes Einzelnen bei. Für weitere Informationen zum Thema SysML sei an dieser Stelle auf folgende Literatur verwiesen [186], [165]. In Tabelle 4-6 sind die Vor- und Nachteile, die sich durch den Einsatz der SysML ergeben, in übersichtlicher Weise dargestellt.

Vorteile	Nachteile
<ul style="list-style-type: none"> – Konsistente Modellierung während der gesamten Systementwicklung – Verbesserung der interdisziplinären Kommunikation – Reduziert Missverständnisse – Erhöht das Gesamtsystemverständnis – Verbessert die Nachvollziehbarkeit 	<ul style="list-style-type: none"> – Ressourceneinsatz während der Einführung – Doppelmodellierung für zeitkontinuierliche Simulation (in Zukunft obsolet) – Abstrakte Systemdarstellung

Tabelle 4-6 Vor- und Nachteile einer standardisierten Modellierungssprache

Einen Vorteil der SysML bildet die kontinuierliche Verwendung während aller Entwicklungsphasen. Sie eignet sich für die Ermittlung von Anforderungen in der frühen Entwicklungsphase, des Weiteren kann sie für die Modellierung des Systemkontextes eingesetzt werden. Darunter ist das Wechselspiel des Systems mit seiner Umgebung zu verstehen. Das Systemkontextmodell stellt die unmittelbare Umgebung des Systems dar. Mithilfe dieses Modells werden erste Aussagen über die Systeminteraktionen mit der Umwelt ermittelt. SysML bietet die Möglichkeit, einzelne Anwendungsfälle, also Dienstleistungen des Systems und somit die

4.3 Standardisierte Modellierung

zentralen Anforderungen, zu modellieren. Die Detailtiefe der Modelle ist skalierbar und kann an den jeweiligen Entwicklungsstand angepasst werden. Durch den Einsatz der Zustandsdiagramme ist es möglich, ereignisorientierte Simulationen durchzuführen. Dies sind nur einige Anwendungsbereiche der SysML im Entwicklungsprozess. Es wird deutlich, dass sich die Nachvollziehbarkeit z. B. von Anforderungen im Systementwicklungsprozess durch den Einsatz der SysML verbessern lässt.

Wie bereits erwähnt, existiert zurzeit keine kommerzielle Lösung für die Anbindung von SysML an zeitkontinuierliche Simulationsprogramme, sodass die Simulation zeitkontinuierlicher Systeme einen zusätzlichen Modellierungsaufwand erfordert. Zum einen muss das Modell mit der spezifischen Modellierungssprache des jeweiligen Simulationsprogramms erstellt werden. Zum anderen muss für die konsistente Entwicklung und die Verbesserung der interdisziplinären Kommunikation das System mit SysML modelliert werden. Wie bereits beschrieben, handelt es sich bei der SysML um eine junge Modellierungssprache, die in den nächsten Jahren auch in zeitkontinuierlichen Simulationsprogrammen zum Einsatz kommen wird. Aus diesem Grund wird dieser Nachteil in Zukunft verschwinden. Für ein Konzept zur durchgängigen Simulation in der Systementwicklung sind diese Umstände zu berücksichtigen, um auch für zukünftige Systementwicklungen einen optimalen Lösungsweg vorzugeben. Für weitere Informationen zu diesem Thema sei auf folgende Quelle verwiesen [186].

Zusammenfassend lässt sich sagen, dass der Einsatz der standardisierten Modellierungssprache SysML die in Kapitel 3 aufgestellten Anforderungen erfüllt. So ist es möglich, heterogene physikalische Systeme zu modellieren. Durch die konsistente Modellierung wird zusätzlich das Gesamtverständnis jedes Einzelnen verbessert. Nicht zuletzt führt die Verwendung von SysML zur Reduzierung der Missverständnisse in der Systementwicklung. Im folgenden Kapitel wird der Zusammenschluss des Simulations-, Modellierungs- und Informationsbereitstellungskonzepts dargestellt.

4.4 Gesamtkonzept

Die durchgängige Simulationsunterstützung bei der Entwicklung von Flugzeugsystemen basiert auf den drei wesentlichen Bausteinen Simulation, Modellierung und Informationsbereitstellung. Diese Bereiche wurden in den vorangegangenen Kapiteln zunächst in Teilkonzepten gestaltet. In diesem Kapitel erfolgt die integrierte Darstellung des Gesamtkonzepts anhand der vier Entwicklungsphasen (Konzept-, Detaillierungs-, Komponentenentwicklungs- und Testphase). Das Konzept beschreibt den Einsatz der Simulationstechnik während der Systementwicklung von der Definition der Hauptsystemanforderungen (M4) bis zur Zertifizierung des Flugzeugs (M12).

4.4.1 Konzeptphase (M4–M5)

Auf Basis der Hauptsystemanforderungen (vgl. Kapitel 4.2.2.1) werden in der Konzeptphase alternative Systemarchitekturen bzw. -konzepte entwickelt. Zunächst erfolgt eine Dekomposition der Hauptsystemanforderungen auf die Teilsysteme, für die anschließend Konzeptalternativen entwickelt werden. Diese werden unabhängig voneinander simuliert, um geeignete Teilsystemkonzepte für die Weiterentwicklung zu ermitteln. Es sei darauf hingewiesen, dass pro Teilsystem durchaus mehrere Konzeptalternativen infrage kommen können. Die Modelldetaillierung in dieser Phase ist gering (Architekturmodelle), sodass ein geringer Ressourceneinsatz erforderlich ist. Die Modelle werden mit den entsprechenden Schnittstellen versehen (siehe Kapitel 4.1.2) und zusammen mit der Modelldokumentation, den Simulationsergebnissen und den Entwicklungsdokumenten in der zentralen Informationsbasis hinterlegt. Durch den Zusammenschluss der Modelle in einer systemübergreifenden Simulation erfolgen eine Analyse der Gesamtkonzeptalternativen und eine Kompatibilitätsprüfung der Teilsysteme. Die Simulation ermöglicht somit die quantitative Überprüfung der übergeordneten Anforderungen. Damit ist bereits im linken Strang des V-Modells eine virtuelle Verifikation des Konzepts, die normalerweise erst nach Beendigung der Bodentests erfolgt (vgl. Abbildung 4-14), möglich. Kommt es zu Inkompatibilitäten zwischen Teilsystemen muss die Konzeptfindung iterativ durchlaufen werden, bis diese Probleme gelöst wurden. Eine erfolgreiche Konzeptüberprüfung resultiert in der Ablage der Systemanforderungsdokumente sowie der Modelle, Simulationsergebnisse und der dazugehörigen Dokumentationen in der Informationsbasis.

Die für die Simulationsdurchführung benötigten Eingangsparameter stehen in der Informationsbasis zur Verfügung und können aus den Anforderungsdokumenten, den Entwicklungsdokumenten vorangegangener Entwicklungsprozesse und vorangegangenen Simulationsstudien entnommen werden. Neben den Daten können in der Informationsbasis Simulationsmodelle früherer Projekte gesucht werden, die sich aufgrund ähnlicher Anforderungen für die Weiterverwendung eignen. Die aktuellen Entwicklungsdaten werden kontinuierlich in der Informationsbasis hinterlegt und stehen somit für die weiteren Entwicklungsphasen zur Verfügung.

Ein Vorteil des Simulationseinsatzes in der Konzeptphase besteht darin, dass verschiedene Konzeptalternativen verglichen werden können und das am besten geeignete Konzept für die Weiterentwicklung ermittelt wird. Der Einsatz der systemübergreifenden Simulationen ermöglicht die frühzeitige Überprüfung von Inkompatibilitäten und beugt damit späten Änderungsschleifen vor.

4.4.2 Detaillierungsphase (M5–M7)

Mit Bestätigung der Systemarchitektur wird die Detaillierung freigegeben. In der Informationsbasis wird die zweite Schicht zur Ablage von Daten dieser Phase aktiviert (siehe Abbildung 4-18). In der Detaillierungsphase werden Simulationen zur Spezifizierung des Systems bzw. zur Überprüfung der Spezifikation von Teilsystemen durchgeführt. In Analogie zur Konzeptphase werden aus Gründen der Ressourcenschonung bis zur Festlegung der Spezifikation alleinstehende Simulationen eingesetzt. Zu Beginn der Detaillierungsphase werden zunächst funktionale Modelle entwickelt, während gleichzeitig eine Dekomposition der Teilsysteme erfolgt. Der erhöhte Modelldetaillierungsgrad ermöglicht eine genauere Analyse des Teilsystemverhaltens im Vergleich zu den Simulationen in der Konzeptphase. Um den Modellierungsaufwand zu reduzieren ist es sinnvoll, geeignete Modelle aus der Informationsbasis wiederzuverwenden. Eingangsparameter der Simulation können, wie schon in der Konzeptphase, ebenfalls aus der Informationsbasis entnommen werden. Durch den Einsatz der Simulation erfolgt in der frühen Detaillierungsphase eine Optimierung der Teilsysteme. Die Ergebnisse dieser Simulationen werden wiederum in der Informationsbasis abgelegt, sodass mit voranschreitendem Entwicklungsprozess die Datenmenge in der zweiten Schicht der Informationsbasis kontinuierlich steigt.

Die voranschreitende Dekomposition macht eine weitere Modelldetaillierung erforderlich. Neben der Systemsimulation kommt in dieser Phase die gestaltorientierte Simulation zum Einsatz. Zum einen ist es damit möglich, das dynamische Verhalten der Teilsysteme zu analysieren, zum anderen können Interaktionen in einer Flugzeugumgebung untersucht werden. Der Modellierungsaufwand für diese Simulationen erfordert einen erhöhten Ressourceneinsatz. Aus diesem Grund wurde in der funktionalen Simulation eine Teilsystemoptimierung durchgeführt, um so das Risiko einer Mehrfachmodellierung in dieser Phase zu minimieren. Eine zusätzliche Reduzierung des Modellierungsaufwands wird durch die Weiterverwendung (Adaption) der funktionalen Modelle erreicht. Als Eingangsparameter der Systemsimulation auf der Verhaltensebene und der gestaltorientierten Simulation dienen die Ergebnisse der funktionalen Simulation. Damit ist es möglich, die Systeminteraktionen in einer losen Simulationskopplung zu berücksichtigen. Die Simulationsmodelle werden mit den entsprechenden Schnittstellen versehen und zusammen mit den Dokumenten und Daten der bisherigen Entwicklung in der Informationsbasis abgelegt.

In Analogie zur Konzeptphase wird am Ende der Detaillierungsphase eine systemübergreifende Simulation durchgeführt (vgl. Abbildung 4-7). In diesem Fall erfolgen die Validierung und Verifikation anhand der Systemanforderungen. Kommt es zu Systeminkompatibilitäten zwischen den Teilsystemen, ist ein Iterationsschritt erforderlich. Nach erfolgreicher Beendigung der Detaillierungsphase werden die Spezifikationsdokumente erstellt und an die jeweiligen Zulieferer ausgegeben.

4.4.3 Komponentenentwicklung (M7–M8)

Die technische Realisierung der Komponenten wird gemäß der Vertragsvereinbarung auf Basis der Spezifikation vom Zulieferer in der Phase der Komponentenentwicklung durchgeführt. Der Zulieferer ist dazu verpflichtet, zusätzlich zur Hardwarekomponente ein Simulationsmodell auf Komponentenebene zu liefern. Dieses Modell ist gemäß der durch die Koordinationsfunktion definierten Regeln zu erstellen. Die Komponentenmodelle können zum einen in folgenden Entwicklungsprozessen eingesetzt werden, zum anderen ist es möglich,

4.4 Gesamtkonzept

die Modelle in HIL-Simulationen in der Testphase einzusetzen, um so z. B. Überlastfälle zu analysieren. Zur Verbesserung der Komponentenentwicklung beim Zulieferer kann dieser ebenfalls auf die Informationsbasis zugreifen. Hierbei ist darauf zu achten, dass nur die Informationen zur Verfügung gestellt werden, die im direkten Zusammenhang mit der Komponentenentwicklung beim Zulieferer stehen.

Parallel zur Komponentenentwicklung (M7–M8) wird in der Testabteilung die Testvorrichtung erstellt und verifiziert. Hierfür greift die Testabteilung auf die relevanten Modelle aus den vorangegangenen Entwicklungsphasen zurück. Dadurch, dass die Modelle gemäß einer Schnittstellenspezifikation in der Informationsbasis hinterlegt sind, können sie von der Testabteilung direkt übernommen und in die Testvorrichtung eingebunden werden. Voraussetzung hierfür ist die Echtzeitfähigkeit der Modelle. Beim Eintreffen der Hardwarekomponenten in der Testabteilung ist die Vorrichtung bereits erstellt und verifiziert, sodass die reale Hardware direkt eingebunden werden kann.

4.4.4 Testphase (M8–M11)

Im Rahmen der Testphase erfolgt eine Verifikation der Geräte (Komponenten) und Systeme anhand der technischen Spezifikationen. Im weiteren Verlauf erfolgen die Funktionstests. In den Integrationstests werden die Systeminteraktionen überprüft und anhand der Systemanforderungen verifiziert (vgl. Abbildung 2-5). In den anschließenden Boden- und Flugtests ist der Einsatz der Simulation nicht mehr erforderlich. Für diese Tests steht die reale Flugzeugumgebung zur Verfügung, sodass keine Umgebungssimulation erforderlich ist. Zudem sind alle Hardwarekomponenten integriert, sodass die Systemgrenze lediglich durch die Flugzeugumgebung gebildet wird und somit keine Simulation von Komponenten, Teilsystemen oder Systemen benötigt wird.

In der Testphase werden Simulationstechniken aus verschiedenen Gründen eingesetzt. So ermöglichen Simulationstechniken es, das Verhalten von Komponenten, Teilsystemen oder Systemen in kritischen Betriebsbereichen zu analysieren, ohne die Hardware zu gefährden. Es kommt durchaus vor, dass einzelne Systemkomponenten nicht rechtzeitig fertiggestellt werden. Durch den Einsatz von Softwaremodellen in HIL-Simulationen ist es möglich, diese Komponenten im Verbund mit realer Hardware zu testen, um so das übergeordnete Systemverhalten analysieren zu können und Verzögerungen im Entwicklungsprozess zu vermeiden. Des Weiteren werden in der Testphase bis einschließlich zu den Integrationstests Umgebungsmodelle für die Bereitstellung der Eingangsparameter an den Systemgrenzen benötigt. Für den frühzeitigen Aufbau sowie die Verifikation der Testvorrichtungen dienen Verhaltensmodelle. Dadurch ist es möglich, die Testvorrichtungen parallel zur Komponentenentwicklung aufzubauen und Komponenten der Zulieferer direkt nach Fertigstellung in die Vorrichtung zu integrieren. Die benötigten Daten, Modelle und Dokumente werden wie schon in den vorangegangenen Phasen aus der Informationsbasis entnommen.

In den Geräte- und Systemtests werden Verhaltensmodelle aus der vorangegangenen Detaillierungsphase sowie Komponentenmodelle der Zulieferer verwendet. Entspricht das System der Spezifikation, wird diese Testphase erfolgreich abgeschlossen. Anschließend werden die Testvorrichtungen modifiziert, sodass mehrere Systeme zusammen in den Funktions- und Integrationstests auf ihre Schnittstellen und Funktion hin überprüft werden. In den Funktions- und Integrationstests kommen funktionale und Architekturmodelle aus der Konzept- und Detaillierungsphase zum Einsatz. Werden im Rahmen der Testphase Geräte-, System-, Funktions- oder Integrationsfehler festgestellt, kommt es zu Rücksprüngen im V-Modell. Je später diese Fehler entdeckt werden, umso kosten- und zeitintensiver ist ihre Behebung. Durch die virtuelle Verifizierung im linken Strang des V-Modells sowie die systemübergreifende Abstimmung in der Entwicklung reduziert sich das Risiko später Änderungen im Gegensatz zum derzeitigen Prozess deutlich. Verlaufen die Tests, einschließlich der Integrationstests, erfolgreich, sind die Teilsysteme endgültig verifiziert. In den anschließenden Boden- und Flugtests erfolgt zunächst eine Verifikation anhand der Hauptsystemanforderungen und später anhand der Flugzeuganforderungen. Den Abschluss der Flugzeugentwicklung bildet die Validierung des Flugzeugs auf Basis der Hauptflugzeuganforderungen.

Zusammenfassend lässt sich sagen, dass das Gesamtkonzept für die durchgängige Simulationsunterstützung im Entwicklungsprozess von Flugzeugsystemen die in Kapitel 3 aufgestellten Anforderungen erfüllt. Das in Kapitel 4.1.2 vorgestellte Simulationskonzept beschreibt den Simulationseinsatz von der Konzeptphase bis zu den Integrationstests. Es verbessert die Entscheidungsqualität in der Konzeptphase und reduziert durch den Einsatz

Entwicklung eines Konzepts für den durchgängigen Einsatz von Simulation

systemübergreifender Simulationen Inkompatibilitäten. Zudem ermöglicht es die Parallelisierung von Aktivitäten und führt damit zur Reduzierung der Entwicklungszeit. Bedingt durch die virtuelle Verifikation in linken Strang des V-Modells, werden Entwicklungsfehler frühzeitig erkannt. Die Verwendung der in Kapitel 4.2.2 vorgestellten Informationsbasis ermöglicht die zentrale Bereitstellung sämtlicher Entwicklungsdaten. Zudem baut sie auf einer bestehenden Infrastruktur auf und ermöglicht eine einfache Erweiterbarkeit. Die individuelle Rechtevergabe sowie die strukturierte Ablage von Daten in der Informationsbasis unterstützen die Datenbeschaffung und verhindern gleichzeitig eine Datenüberflutung des Nutzers. Über Regeln und Standards wird der Umgang mit der Informationsbasis zusätzlich benutzerfreundlicher. Durch die Verwendung der einheitlichen grafischen Modellierungssprache SysML ist es möglich, den Entwicklungsprozess von der Definition der Anforderungen über die zeitdiskrete Simulation mit Zustandsautomaten bis hin zur physikalischen Modellierung von zeitkontinuierlichen Systemen zu unterstützen. Zusätzlich erhöht eine einheitliche Modellierung die Kommunikation in der Entwicklung und verbessert damit das Gesamtsystemverständnis aller Beteiligten.

Die Einführung eines Informationsmanagement, einer Methodik oder einer Modellierungssprache verursacht Investitions- und Schulungskosten der Benutzer. Aus diesem Grund wird bei der Einführung des Konzepts für die durchgängige Simulationsunterstützung im Entwicklungsprozess von Flugzeugsystemen zunächst der Ressourcenaufwand steigen. Durch die beschriebenen Vorteile des Konzepts wird sich dieser zusätzliche Aufwand innerhalb kurzer Zeit amortisieren.

5 Gesamtkonzeptevaluierung

Ausgehend von einer Analyse der Schwachstellen des derzeitigen Flugzeugsystementwicklungsprozesses wurden in Kapitel 3 Anforderungen an ein zukünftiges Konzept zum durchgängigen Einsatz von Simulationstechniken in der Systementwicklung aufgestellt. Im anschließenden Kapitel 4 wurde, basierend auf diesen Anforderungen, ein generisches Konzept für den Entwicklungsprozess von Flugzeugsystemen entworfen. In diesem Kapitel erfolgt die Umsetzung des vorgestellten Konzepts. Zusätzlich zu den im vorangegangenen Kapitel beschriebenen qualitativen Vorteilen wird der Nutzen des Konzepts anhand von Anwendungsbeispielen quantifiziert. Es werden technische Lösungen für das Simulationskonzept und die Informationsbereitstellung beschrieben. Diese Lösungen stellen Möglichkeiten für die Realisierung des Konzepts dar, sollen aber nicht als einzige Möglichkeit der Umsetzung betrachtet werden. Das in Kapitel 4 vorgestellte Konzept stellt eine generische Spezifikation dar, für die verschiedene Realisierungen möglich sind.

Im Folgenden wird die Umsetzung des Konzepts anhand einer Brennstoffzellensystementwicklung durchgeführt. Aus diesem Grund werden zunächst die Grundlagen sowie die Strategien der Automobil-, Schiffsbau- und Flugzeugindustrie in Bezug auf die Brennstoffzellentechnologie vorgestellt.

5.1 Grundlagen von Brennstoffzellensystemen

Vor dem Hintergrund der globalen Erwärmung gewinnen regenerative, umweltfreundliche und emissionsarme Technologien zur Energieversorgung immer mehr an Bedeutung. Die Brennstoffzelle stellt eine effiziente Alternative für die Bereitstellung elektrischer Energie in mobilen und stationären Applikationen dar [97].

Das Prinzip der direkten Umwandlung von chemischer in elektrische Energie ohne den Umweg über die Verbrennung ist seit 1839 bekannt, als W. Grove die erste Brennstoffzelle entwickelte [99]. Die grundsätzliche Funktionsweise einer Brennstoffzelle ist bei allen Varianten die gleiche. Ein Brennstoff wird in einer chemischen Reaktion oxidiert, hierdurch wird Wärme abgegeben und zugleich elektrische Energie erzeugt, d. h., die Oxidationsreaktion von Wasserstoff und Sauerstoff zu Wasser verläuft exotherm.

Die Umwandlung von chemischer in elektrische Energie erfolgt in der Brennstoffzelle direkt. Daraus resultieren höhere Wirkungsgrade als in herkömmlichen Systemen zur Stromerzeugung, in denen eine mehrmalige Energiewandlung stattfindet. Ein weiterer Vorteil der Brennstoffzellentechnologie liegt in der Verminderung der Schadstoffemission. Werden Brennstoffzellen mit reinem Wasserstoff betrieben, so werden keine Schadstoffe ausgestoßen, das Reaktionsprodukt ist reines Wasser.

Im folgenden Abschnitt wird stellvertretend die Funktionsweise einer Protonenaustauschmembran-Brennstoffzelle (Abk. PEMFC; engl.: Proton Exchange Membrane Fuel Cell) beschrieben. In der folgenden Abbildung 5-1 ist der prinzipielle Aufbau einer PEMFC dargestellt.

An der Anode der PEM-Brennstoffzelle wird Wasserstoff zugeführt und in einer katalytischen Reaktion ionisiert. Die hierbei frei werdenden Elektronen sammeln sich an der Anode und führen dort zu einer Absenkung des Potentials. Die Wasserstoffionen wandern durch die protonenleitende Membran zur Kathode und reagieren mit den Sauerstoffmolekülen. Gleichzeitig fließen, bedingt durch den Potentialunterschied zwischen Anode und Kathode, über einen externen Stromkreis Elektronen zur Kathode. Für die gesamte PEM-Brennstoffzelle ergibt sich somit folgende Reaktionsgleichung.



Hierbei gibt die Standardreaktionsenthalpie Δh_R die bei der exothermen Reaktion freiwerdende Energie an. Sie kann über die Standardbildungsenthalpien der Edukte und Produkte ermittelt werden und ergibt sich im Fall von gasförmigen Produktwasser zu 241,83 kJ/mol. Die folgende Abbildung zeigt eine exemplarische Polarisationskennlinie einer PEM-Brennstoffzelle.

Gesamtkonzeptevaluierung

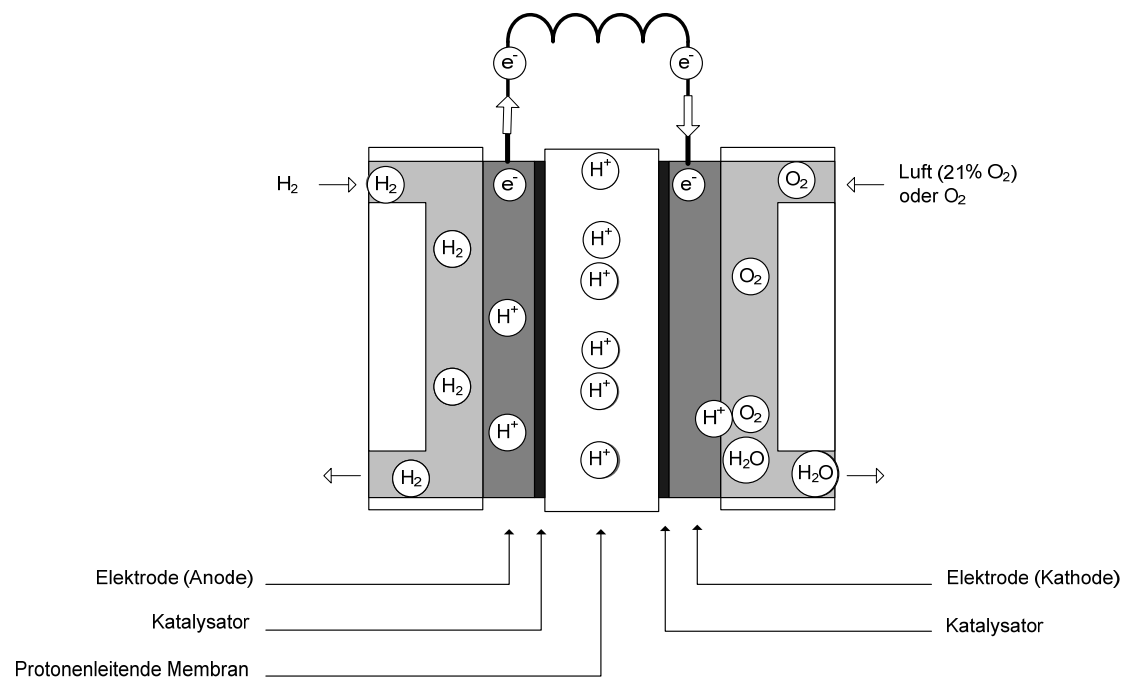


Abbildung 5-1 Prinzipdarstellung einer PEMFC

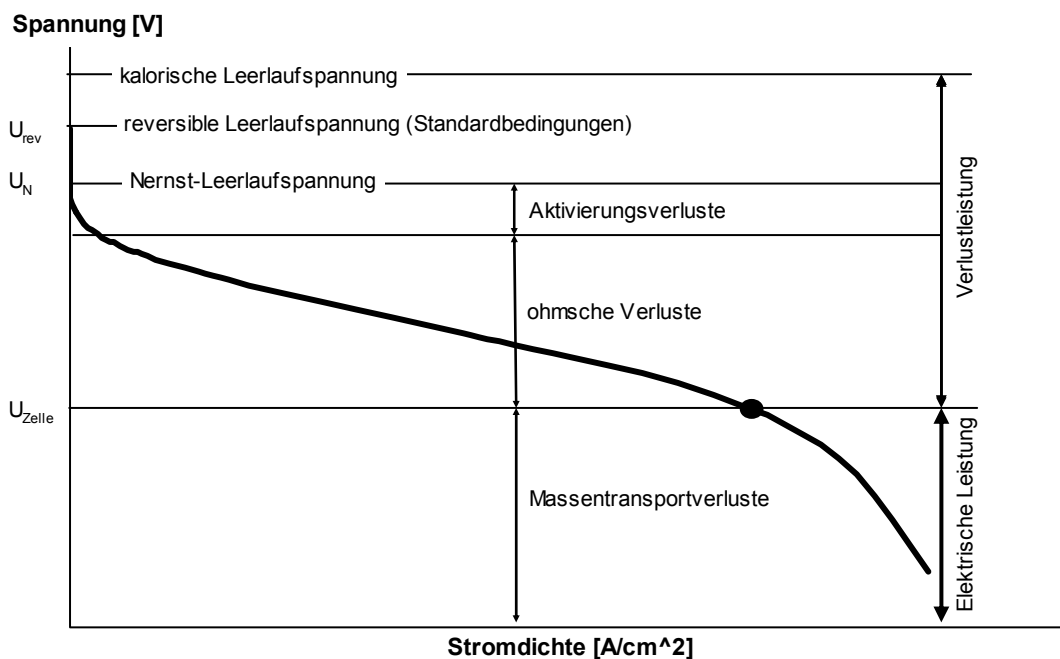


Abbildung 5-2 Kennlinie einer Brennstoffzelle

Die Brennstoffzellenkennlinie wird von drei Effekten maßgeblich beeinflusst. Im Bereich kleiner Stromdichten überwiegen die Aktivierungsverluste. Diese Aktivierungsverluste entstehen dadurch, dass sich an der Grenzschicht zwischen den Elektroden und dem Elektrolyten ein elektrisches Feld aufbaut, welches die Ladungsträger am Fluss hindert. Ein steigender Ladungsträgerfluss wirkt der Ausbildung des Feldes entgegen. Aus diesem Grund sind die Aktivierungsverluste nur im Bereich kleiner Stromdichten von Bedeutung [168]. Im weiteren Verlauf der Kennlinie, also mit steigender Stromdichte, dominieren die ohmschen Verluste. Sie entstehen an allen ohmschen Widerstandsanteilen, die den Ladungsträgerfluss im Stromkreis hindern, und sind durch einen linearen Zusammenhang zwischen Spannung und Strom bzw. Stromdichte charakterisiert. Die ohmschen Verluste treten im gesamten Verlauf der

5.1 Grundlagen von Brennstoffzellensystemen

Kennlinie auf, machen sich aber im Bereich sehr großer Stromdichten bis hin zum Erreichen des völligen Spannungszusammenbruchs nicht mehr bemerkbar, da sie hier von den in diesem Bereich einsetzenden Massentransportverlusten überlagert werden. Die Massentransportverluste begrenzen die maximale Stromstärke und damit auch die Stromdichte der Brennstoffzelle. Sie beruhen auf Diffusions- sowie Konvektionsverlusten bei Zuleitung und Abtransport der Reaktionspartner an Anode und Kathode [90].

Unter Anwendung der Butler-Volmer-Gleichung kann die Durchtrittsspannung und damit die Brennstoffzellenspannung in Abhängigkeit der Stromdichte errechnet werden [99].

$$U(i) = U_{rev} - i \cdot r - A \cdot \ln \left(\frac{i + i_n}{i_0} \right) - m \cdot e^{(n \cdot i)} \quad (5-2)$$

Dabei bezeichnet U die Spannung der Zelle, i die Stromdichte in der Membran und U_{rev} die reversible Leerlaufspannung der Zelle, die mithilfe der Gibbsschen Energie für Standardbedingungen ermittelt werden kann. Die Gibbssche Energie gibt hierbei den Teil der frei werdenden Energie an, der maximal in elektrische Energie umgewandelt werden kann. Die einzelnen Terme dieser Gleichung bilden jeweils einen der drei beschriebenen Effekte der Brennstoffzelle ab.

Umgebungsbedingungen wie Druck, Temperatur und Luftfeuchtigkeit beeinflussen das Betriebsverhalten der Brennstoffzelle. Mit steigender Betriebstemperatur überlagern sich zwei Effekte. Zum einen sinkt der Membranwiderstand, zum anderen sinkt die reversible Zellspannung. Diese beiden Effekte wirken einander entgegen, sodass die Änderung der Brennstoffzellenspannung gering ist [105]. Eine Variation des Betriebsdrucks mit positivem Gradienten führt zu einem Anstieg der Zellspannung und damit zu einer Erhöhung der Leistung. Dieses Verhalten ist auf die steigende Konzentration sowohl der Sauerstoff- als auch der Wasserstoffmoleküle zurückzuführen. Bei steigender Konzentration stehen mehr Ladungsträger pro Volumen zur Verfügung, dies hat eine Steigerung der Ladungsträgerdichte sowohl auf der Oberfläche der Elektrode als auch in der Membran zur Folge. Der Einfluss der Partialdrücke auf die reversible Zellspannung wird in der Nernstschen Gleichung berücksichtigt [99].

Neben der PEM-Brennstoffzelle existieren weitere Brennstoffzellentypen, die je nach Betriebstemperatur in Niedertemperatur- und Hochtemperaturbrennstoffzellen unterschieden werden. Für das weitere Verständnis sind Kenntnisse der verschiedenen Brennstoffzellen nicht erforderlich. Für den interessierten Leser sei an dieser Stelle auf folgende Literatur verwiesen [33], [90], [99], [171].

Eine Herausforderung bei der Nutzung von Brennstoffzellen in mobilen Applikationen ist die Lagerung des Treibstoffs. Gasförmiger Wasserstoff zeichnet sich durch eine niedrige Energiedichte aus. Für den Einsatz von Wasserstoff in mobilen Anwendungen ist es jedoch notwendig eine hohe Energiespeicherdichte zu erzielen, um eine geeignete Reichweite des Fahrzeugs bei geringem Tankvolumen zu gewährleisten. Ein zusätzliches Gefahrenpotenzial birgt die hohe Explosivität bei der Reaktion von frei werdendem Wasserstoff mit Sauerstoff, der sogenannten Knallgasreaktion. Für die Speicherung von Wasserstoff haben sich die Druckspeicherung, die kryogene Speicher und die Metallhydridspeicherung etabliert. Die folgende Tabelle zeigt die wichtigsten Eigenschaften dieser drei Methoden zur Wasserstoffspeicherung bezogen auf eine einheitliche Vergleichsmenge von 50 Litern Benzin. Die Masse ist dabei auf den Tank inklusive Brennstoff bezogen.

Alle drei Technologien zeigen individuelle Vor- und Nachteile. Die kryogene Wasserstoffspeicherung zeichnet sich durch eine geringe Masse und ein im Vergleich zu den anderen Speichermethoden geringes Volumen aus, allerdings ist die niedrige Temperatur von -253°C mit einem großen Energie- und Isolationsaufwand verbunden.

Der Wasserstoffdruckspeicher erfordert einen hohen Druck von 250 bar und liegt mit seiner Masse im Mittelfeld der drei Speicherarten, bietet allerdings keine hohe Energiedichte, wodurch sich ein großes Volumen ergibt. Mittlerweile werden Druckspeicher mit bis zu 650 bar druckbeaufschlagt, wodurch sich die Energiedichte und damit das Volumen und die Masse des Speichers verringern.

	Wasserstoff- druckspeicher	Flüssigwasser- stoffspeicher	Metallhydrid- Wasserstoff- speicher	Benzin- tank (50 Liter)
Temperatur [°C]	20	-253	k. a.	< 200
Druck [Pa]	250e5	1e5–5e5	1e5 ... 50e5	~ 1e5
Masse [kg]	200	120	1300	~ 52
Volumen [l]	740	360	170	~62
Eigenschaften	Verbund- werkstoffe	Doppelwandiger Tank mit Vakuum- Superisolation	Absorption von Wasserstoff an Metall	Vergleichs- tank

Tabelle 5-1 Eigenschaften von Wasserstoffspeichern bezogen auf eine 50 Liter Benzintank [105]

Der Metallhydridspeicher, bei dem der Wasserstoff an Metallen zu Metallhydrid gebunden ist, bietet im Vergleich das geringste Volumen, allerdings ist diese Methode mit einem extrem hohen Gewicht für Anwendungen in der Luftfahrt uninteressant. Zusätzlich wird für die reversible Reaktion Wärme benötigt, um den Wasserstoff freizusetzen. Im Vergleich zum Benzintank wird die Überlegenheit der Speicherung in Form von Kohlenwasserstoffen deutlich.

Brenngasreformierung

Für den wirtschaftlichen Einsatz von Brennstoffzellen in der Energiewirtschaft ist es erforderlich, dass verfügbare Brennstoffe direkt in den Brennstoffzellensystemen eingesetzt werden können [189]. Gründe hierfür sind die bereits angesprochene hohe Energiedichte konventioneller Brennstoffe wie Benzin oder Diesel sowie die Nutzung der bestehenden Infrastruktur. Ein weiterer Grund besteht darin, dass Wasserstoff in der Natur nicht in ungebundener Form vorkommt und für dessen Herstellung entweder die Reformierung von Wasserstoff aus fossilen Energieträgern oder Elektrolyseprozesse benötigt werden. Aus diesem Grund werden im Folgenden verschiedene Reformierungsmethoden zur Wasserstoffgewinnung aus Kohlenwasserstoffen beschrieben.

Die Dampfreformierung (Abk. STR) gehört zu den am höchsten entwickelten und häufig eingesetzten Technologien zur Wasserstoffproduktion aus Kohlenwasserstoffen. Es handelt sich hierbei um eine katalytische Reformierungsreaktion [90]. Dem zu reformierenden Kohlenwasserstoff wird Wasserdampf in der Regel in einem überstöchiometrischen Verhältnis hinzugefügt, erhitzt und dem Katalysator zugeführt. Unter Verwendung von Kohlenwasserstoffen ist der Verlauf der Dampfreformierung endotherm, allerdings kann diese Eigenschaft in Abhängigkeit vom verwendeten Edukt und anderen Operationsparametern, wie Druck und Temperatur, verstärkt, aber auch bis hin zu exothermem Verhalten verschoben werden. Typischerweise wird die Dampfreformierung von Kohlenwasserstoffen bei hohen Temperaturen und niedrigen Drücken betrieben, da hierdurch der optimale Prozentsatz der Stoffumwandlung erreicht werden kann [82], [99]. Spezifische Vorteile dieser Reformierungsmethode liegen in der vergleichsweise hohen Wasserstoffausbeute im Vergleich zu anderen Methoden. Nachteilig ist neben der hohen Betriebstemperatur, die einen hohen Energieaufwand zur Erwärmung der Eingangsstoffe nötig macht, die Produktion von Kohlenmonoxid. Sie stellt für alle Vertreter der Niedertemperaturbrennstoffzellen ein Problem dar, da diese zum Teil bereits auf geringe Mengen von 10 bis 20 ppm mit deutlichem Leistungsabfall reagieren.

Ein weiteres in der Praxis genutztes Reformierungsverfahren stellt die partielle Oxidation (Abk. POX) dar. Bei diesem Verfahren wird dem Kohlenwasserstoff Sauerstoff bzw. Luft zugemischt. Die partielle Oxidation ist exotherm und kann mit oder ohne Katalysator betrieben werden. Im Falle einer katalytischen POX liegt die Betriebstemperatur um 700 °C. Im Temperaturbereich von etwas 1100 bis 1500 °C wird aufgrund der hohen Temperatur kein Katalysator benötigt, dies hat eine erhöhte Schwefelresistenz zur Folge. Aufgrund der exothermen Reformierungsreaktion benötigt die POX im Gegensatz zur Dampfreformierung nach einer Startphase keine weitere Wärmezufuhr. Als Nachteile sind die geringe Wasserstoffproduktion sowie der Kohlenmonoxidanteil im Reformatgas zu nennen [33].

Die autotherme Reformierung (Abk. ATR) stellt eine weitere Methode zur Wasserstoffgewinnung dar. Sie stellt eine Kombination aus partieller Oxidation und Dampfreformierung dar. Mithilfe der Wasserdampf-zu-Kohlenstoff- (Abk. S/C) und Sauerstoff-zu-Kohlenstoffverhältnisse (O_2/C) ist es möglich, die abgegebene Wärme der exothermen POX

5.1 Grundlagen von Brennstoffzellensystemen

so einzustellen, dass sie den Bedarf an Wärmezufuhr der STR deckt. Die Gesamtreaktion ist damit autotherm, es wird also weder Wärme zu- noch abgeführt. Ein weiterer Vorteil dieser Methode ist die im Vergleich zur POX erhöhte Wasserstoffproduktion. Wie in den vorangegangenen Methoden enthält das Reformatgas der ATR ebenfalls Kohlenmonoxid. Für den Betrieb von Niedertemperaturbrennstoffzellen ist aus diesem Grund eine Gasreinigung z. B. in einer Wasser-Gas-Shift-Reaktion mit anschließender Feingasreinigung mithilfe einer Palladium-Silber-Membran nötig. Zudem resultiert aus dem komplexen Systemaufbau eine signifikante Erhöhung der Regelung.

Neben den beschriebenen Standardreformierungsmethoden wurden im Jahr 2005 die Patentrechte des Unternehmens EADS IW für die Erfindung einer neuen Reformierungsmethode anerkannt [71]. Es handelt sich hierbei um die partielle Dehydrogenierung von Kohlenwasserstoffen (Abk. PDh). Das Verfahren der partiellen Dehydrogenierung besitzt in der chemischen Industrie zwar ein durchaus vielfältiges Anwendungsgebiet, jedoch ist der entstehende Wasserstoff hierbei in der Regel nur ein Abfallprodukt. Es zur Produktion von Wasserstoff einzusetzen ist ein neuer Denkansatz, von dem man sich in der Zukunft große Vorteile im Vergleich zu den herkömmlichen Verfahren zur Wasserstoffgewinnung erhofft. Der Aufbau eines PDh-Reformers ist simpel. Er besteht aus einer Reaktionskammer, in der ein Katalysator enthalten ist, und einem Separator. In die Reaktionskammer wird ein langkettiger Kohlenwasserstoff, wie z. B. Kerosin, eingeleitet. Bei einer Temperatur zwischen 200 bis 400 °C und einem Druck zwischen 3 bis 20 bar erfolgt die Wasserstoffabspaltung am Katalysator. Als Produkte dieser Reaktion entstehen dehydrierter Kohlenwasserstoff und Wasserstoff. Anschließend werden die entstandenen Produkte im Separator getrennt. Im Gegensatz zu den bereits vorgestellten Methoden erfolgt bei der PDh keine Oxidationsreaktion, d. h., es entsteht reiner Wasserstoff frei von Kohlenmonoxidanteilen. Der dehydrierte Kohlenwasserstoff kann weiterhin z. B. in Verbrennungsmotoren eingesetzt werden. Allerdings sei an dieser Stelle darauf hingewiesen, dass die Abspaltung des Wasserstoffs einen geringeren Heizwert des dehydrierten Kohlenwasserstoffs zur Folge hat. In der nachfolgenden Abbildung ist die Reaktionsgleichung des PDh-Verfahrens dargestellt, wobei als Edukt $C_{11}H_{24}$ (Undekan) verwendet wurde [33].

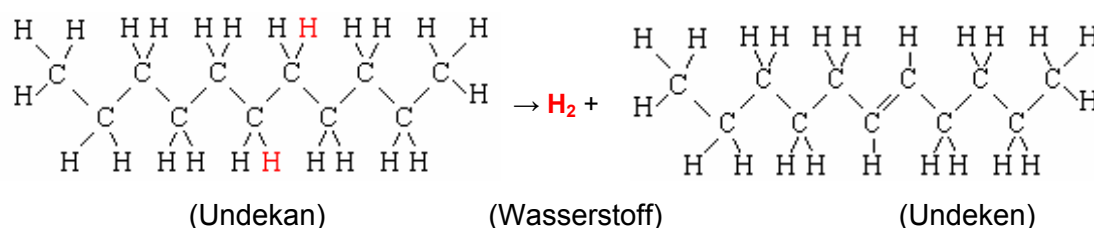


Abbildung 5-3 Reaktionsgleichung der partiellen Dehydrogenierung von Undekan

Die dazugehörige allgemeine chemische Reaktionsgleichung lautet:



Die Reaktion kann grundsätzlich sowohl in der flüssigen als auch in der gasförmigen Phase ablaufen. Für den Betrieb mit Undekan im flüssigen Aggregatzustand ergibt sich die Standardreaktionsenthalpie zu 137,5 kJ/mol und für den gasförmigen Aggregatzustand zu 125,5 kJ/mol. Damit läuft die Reaktion endotherm ab, für ihre Aufrechterhaltung muss somit Energie in Form von Wärme zugeführt werden. Aus der Reaktionsgleichung wird deutlich, dass nur ein Molekül Wasserstoff pro Kohlenwasserstoffmolekül abgespalten wird. Damit ist die Wasserstoffproduktion der PDh von den bisher vorgestellten Methoden am niedrigsten (siehe Tabelle 5-2). Dieser Effekt wird durch die derzeitige Kohlenwasserstoffumsetzung von 15 % noch verstärkt. Zum gegenwärtigen Zeitpunkt befindet sich die Entwicklung dieser Technologie in der Laborerprobungsphase, für die Zukunft ist mit einer signifikanten Erhöhung der Umsetzung zu rechnen. Durch den Einsatz geeigneter Katalysatoren sowie durch die Optimierung der Reaktionsfläche erhofft man sich die Abspaltung mehrerer Moleküle Wasserstoff pro Kohlenwasserstoffmolekül und somit eine weitere Steigerung der Wasserstoffproduktion. Die spezifischen Vorteile dieser Technologie im Vergleich zu anderen Reformierungsverfahren liegen im vereinfachten resultierenden Gesamtsystem, bedingt durch einfachen Reformeraufbau sowie den Wegfall einer Gasreinigung. Im Gegensatz zu den

Gesamtkonzeptevaluierung

herkömmlichen Reformierungsmethoden entstehen bei der PDh-Reformierung keine Treibhausgase. Ein weiterer Vorteil der PDh besteht darin, dass je nach verwendetem Katalysator eine Entschwefelung des Kohlenwasserstoffes überflüssig ist. Für eine quantitative Bewertung der PDh werden in Kapitel 5.2.1 verschiedene Brennstoffzellensystemkonzepte simuliert. Unter anderen wird ein Brennstoffzellen-PDh-System mit konventionellen Flugzeugsystemen zur Bereitstellung elektrischer Energie verglichen.

In der nachfolgenden Tabelle 5-2 sind die wichtigsten Eigenschaften der vorgestellten Reformierungsmethoden noch einmal übersichtlich dargestellt.

	STR	POX	ATR	PDh
Wasserstoffproduktion (CnHm)	Hoch (n+m/2)H ₂	Niedrig (m/2)H ₂	Mittel (m/2+n-2*n*O ₂ /C)H ₂	Sehr niedrig H ₂
Betriebs- temperatur	800–1100 °C	600–900 °C 1100–1500 °C ohne Katalysator	600–900 °C	200–400 °C
Prozess	5 Schritte: Verdampfen Mischen Reformieren Gas- und Feingas- reinigung	5 Schritte: Verdampfen Mischen Reformieren Gas- und Feingas- reinigung	5 Schritte: Verdampfen Mischen Reformieren Gas- und Feingas- reinigung	3 Schritte: Erwärmen Dehydrogenierung Separation
Sonstiges	Endotherm - Reformatgas- reinigung - Ent- schwefelung	Exotherm - Reformatgas- reinigung - Ent- schwefelung	Autotherm - Reformatgas- reinigung - Ent- schwefelung - Hohe Komplexität	Endotherm - kein CO, CO ₂ , N ₂ im Produktgas - keine Entschwefel- lung

Tabelle 5-2 Vergleich verschiedener Reformierungsmethoden

Die Wasserproduktion ist bei den einzelnen Methoden auf einen beliebigen Kohlenwasserstoff bezogen. Für die Wasserproduktion der autothermen Reformierung ist das Sauerstoff-zu-Kohlenstoff-Verhältnis und damit die Reaktion der partiellen Oxidation ausschlaggebend. Neben den vorgestellten Reformierungsmethoden existieren weitere Verfahren, wie z. B. die Plasmareformierung, die aufgrund des hohen Leistungsbedarfs sowie des niedrigen Entwicklungsstands für den Einsatz in mobilen Applikationen ungeeignet sind. Für weitere Informationen zur Reformierung von Kohlenwasserstoffen sei an dieser Stelle auf folgende Literaturquellen verwiesen [33], [90], [82].

Im folgenden Kapitel wird die Bedeutung von Wasserstoff als Energieträger der Zukunft beschrieben. Des Weiteren wird vor diesem Hintergrund der Einfluss der Brennstoffzellentechnologie in den Bereichen Luftfahrt-, Schifffahrt-, Raumfahrt- und Automobilindustrie dargestellt.

Bedeutung der Brennstoffzellentechnologie für mobile Anwendungen

Ein Großteil der Weltenergieversorgung basiert auf fossilen Brennstoffen. Nach wie vor ist Erdöl der wichtigste Energielieferant der Welt. Rund 40 % der benötigten Energie wird aus Erdöl bezogen. Bedingt durch die Endlichkeit der fossilen Brennstoffe, erfolgt eine stetige Verteuerung dieser Energieträger. Insbesondere die Wirtschaft von Ländern wie z. B. Deutschland, die über geringe Ressourcen im Bereich der fossilen Brennstoffe verfügen und auf deren Import angewiesen sind, wird zukünftig von dieser Entwicklung negativ beeinflusst werden. Die Einflussnahme der Öl fördernden Länder (OPEC-Staaten) auf die Weltwirtschaft wurde in den vergangenen Jahren immer deutlicher. Vor diesem Hintergrund ist ein Umdenken im Bereich der zukünftigen Energieversorgung für das wirtschaftliche Überleben von großer Bedeutung. Der Einsatz regenerativer Energien, wie Solarenergie, Energie aus Biomasse, Windenergie und Wasserstoff als Energieträger, bietet eine alternative Energieversorgung und ermöglicht damit die Autonomie von fossilen Energielieferanten. Von dieser Entwicklung sind insbesondere die Schifffahrt-, Luftfahrt- und Automobilindustrie betroffen. Als Folge dieser veränderten Ausgangssituation müssen alternative Antriebskonzepte

5.1 Grundlagen von Brennstoffzellensystemen

entwickelt werden. Aus diesem Grund wird die Brennstoffzellentechnologie als eine effiziente Alternative zu den konventionellen Verbrennungsmotoren angesehen.

Die erste Anwendung der Brennstoffzellentechnologie erfolgte in den 50er-Jahren des letzten Jahrhunderts in der bemannten Raumfahrt [101]. Im Space Shuttle werden derzeit drei alkalische Brennstoffzellen für eine kontinuierliche elektrische Energieversorgung von 7 kW pro Brennstoffzellenstapel (engl.: stack) bei einer Lebensdauer von 2000 Stunden eingesetzt [64].

Das Potenzial für den Einsatz von Brennstoffzellensystemen in maritimen Applikationen wurde bereits Anfang der 70er-Jahre erkannt. Im Auftrag des Bundesamts für Wehrtechnik und Beschaffung entwickelte die Howaldtswerke Deutsche Werft AG in Zusammenarbeit mit Ferrostaal und dem Ingenieurkontor Lübeck das U212, das erste brennstoffzellenbetriebene U-Boot der Welt [90]. Der Vorteil eines Niedertemperatur-Brennstoffzellen-Antriebes besteht in der geringen Wärmesignatur sowie der Geräusch- und Vibrationsreduzierung gegenüber konventionellen Antrieben. Mittlerweile kommen Brennstoffzellen als Notstromversorgung (Abk. APU; engl.: Auxiliary Power Unit) auf Frachtern, Containerschiffen und Luxusjachten zum Einsatz.

In den letzten Jahren hat sich das Umweltbewusstsein in der Bevölkerung wesentlich verändert. Begriffe wie globale Erwärmung und Ozonloch haben das Konsumverhalten verändert. Zusätzlich beeinflussen ständig steigende Energiekosten das Verbraucherverhalten dahin gehend, dass bei der Anschaffung eines Produkts Umweltverträglichkeit, Energieverbrauch und Recycling bei der Entscheidungsfindung immer größere Bedeutung zukommen. Dieses veränderte Konsumverhalten wirkt sich besonders stark auf die Automobilindustrie aus, die ihre Produkte an die Anforderungen der Kundenwünsche anpassen muss. Zudem werden weitere Anforderungen von gesetzlicher Seite an die Automobilindustrie gestellt. So hat das Europaparlament für 2007 gesetzliche Maßnahmen angekündigt, die Flottenemission bis 2012 auf 120 g/km zu senken [92]. In Abbildung 5-4 sind die Kohlendioxidemission verschiedener Kraftfahrzeugantriebe für einen durchschnittlichen europäischen Kompaktwagen gemäß der EG-Richtlinie 80/1268/EWG dargestellt. Es wird deutlich, dass mit konventionellen Verbrennungsmotoren die von der EU geforderten CO₂-Werte nicht erreicht werden können. Auch der Hybridantrieb, bestehend aus Elektro- und Dieselmotor, kann die geforderten 120 g/km nicht erreichen.

Als Hybridantrieb bezeichnet man allgemein die Kombination verschiedener Antriebsprinzipien oder verschiedener Energiequellen für eine Antriebsaufgabe innerhalb einer Anwendung. Den ersten Hybridantrieb für ein Kraftfahrzeug, bestehend aus einem Elektro- und einem Verbrennungsmotor, entwickelte Ferdinand Porsche bereits im Jahre 1896 [131]. Heutzutage bestehen die Hybridantriebe in Automobilen entweder aus einer Kombination aus Diesel- und Elektromotor oder Benzin- und Elektromotor. Der Vorteil der Hybridtechnologie besteht darin, dass der Verbrennungsmotor beim Hybridantrieb häufiger und länger in einem optimalen Wirkungsgradbereich betrieben werden kann. Anfallende überschüssige Energie wird über einen Generator zum Laden der Batterien verwendet. Bei Beschleunigungsvorgängen arbeiten Verbrennungs- und Elektromotor gemeinsam. Anfallende Bremsenergie wird ebenfalls der Batterie zurückgeführt. Der Hybridantrieb ermöglicht es, den Verbrennungsmotor unter wesentlich verbrauchsgünstigeren Bedingungen zu betreiben. Insbesondere im Stadtverkehr und beim Bergabfahren trägt die Rückgewinnung zur Verbrauchsminderung bei. Für zukünftige Hybridfahrzeuge ist auch eine Kombination aus Brennstoffzellenantrieb und Verbrennungsmotor denkbar. Für weitere Informationen zum Thema Hybridantriebe sei an dieser Stelle auf folgende Literatur verwiesen [16].

Es ist offensichtlich, dass durch den Einsatz von Brennstoffzellen in Kraftfahrzeugen eine deutliche Reduzierung des Treibhausgases CO₂ erreicht werden kann. Aus diesem Grund sehen die Autohersteller Volkswagen und DaimlerChrysler den Brennstoffzellenantrieb als Schlüsseltechnologie der Zukunft an [113].

Die erste Anwendung einer Brennstoffzelle zum Antrieb in der Automobilindustrie wurde 1966 vorgestellt. Hierbei handelte es sich um einen Kleinbus des Autoherstellers Opel, der aufgrund des installierten Brennstoffzellensystems nur zwei Passagieren Platz bot. Die Brennstoffzelle stellte eine Leistung von 5 kW für eine Reichweite von ca. 200 km zur Verfügung. Die Lebensdauer der Brennstoffzelle betrug damals gerade einmal 1000 Stunden [90].

Mittlerweile wartet nahezu jeder namhafte Automobilhersteller mit mindestens einem brennstoffzellenbetriebenen Fahrzeugkonzept auf. Bei den eingesetzten Brennstoffzellen handelt es sich zumeist um PEM-Brennstoffzellen.

Gesamtkonzeptevaluierung

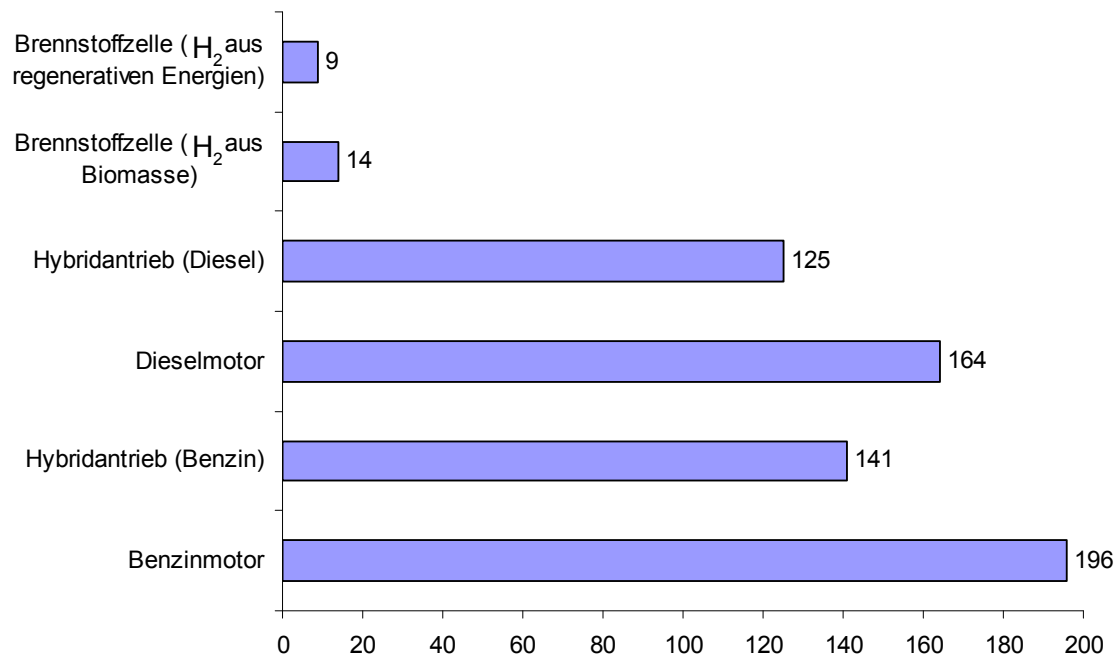


Abbildung 5-4 Kohlendioxidemission verschiedener Antrieben [30]

Der Wasserstoff wird dabei gasförmig in Drucktanks oder flüssig (kryogen) in isolierten Behältern bei einer Temperatur von $-253\text{ }^{\circ}\text{C}$ mitgeführt. Im geringeren Umfang kommen auch Metallhydridspeicher zum Einsatz. DaimlerChrysler stellte auf dem Genfer Automobilsalon 2005 mit der B-Klasse „F-Cell“ die zweite Generation seiner Brennstoffzellenfahrzeuge vor. In Abbildung 5-5 ist dieses Fahrzeug dargestellt, der PEMFC-Stack (blau) liefert eine Leistung von 100 kW. Durch den Einsatz von Wasserstoffdruckspeichern ist eine Reichweite von 400 km möglich. Die Brennstoffzelle ist die Schlüsseltechnologie für das emissionsfreie Fahren der Zukunft und das langfristige Ziel der Zukunftsstrategie von DaimlerChrysler [34].



Abbildung 5-5 Mercedes F-Cell B-Klasse [35]

Neben der Verwendung von Wasserstoff in Brennstoffzellen zur elektrischen Energiebereitstellung gehen einige Automobilhersteller einen anderen Weg. Bei BMW wird der Wasserstoff weiterhin in einem konventionellen Verbrennungsmotor oxidiert. Wie im Falle der brennstoffzellenbetriebenen Fahrzeuge ist die Wasserstoffverbrennung ebenfalls emissionsfrei, allerdings wird der Wirkungsgrad des Systems durch den Carnot-Wirkungsgrad begrenzt und liegt damit unter dem mit einem Brennstoffzellensystem erreichbaren Wirkungsgrad. Der Vorteil besteht darin, dass weiterhin konventionelle Technologien zum Einsatz kommen und somit Entwicklungskosten nur im geringen Maße anfallen.

Die Wasserstoffspeicherung sowie der Aufbau einer Wasserstoffinfrastruktur stellen zum gegenwärtigen Zeitpunkt die größte Herausforderung für die Kommerzialisierung von Brennstoffzellenfahrzeugen dar. Die inadäquate Wasserstoffbereitstellung resultiert in einer erhöhten Anforderung an die Reichweite; dies führt zu einer Vergrößerung der Wasserstoffdruckspeicher und damit zur Reduzierung des freien Fahrzeugvolumens. Aus diesem Grund werden für die Übergangszeit vom Verbrennungsmotor zum brennstoffzellenangetriebenen Elektrofahrzeug Konzepte entwickelt, die eine Nutzung von Kohlenwasserstoffen als Wasser-

5.1 Grundlagen von Brennstoffzellensystemen

stoffspeicher zulassen. Im NECAR-Projekt von DaimlerChrysler wurde 1997 erstmals die Wasserstoffreformierung aus flüssigem Methanol demonstriert [90].

Eine weitere Einsatzmöglichkeit für die Brennstoffzellentechnologie an Bord eines Kraftfahrzeugs bietet die Sekundärenergieversorgung. In den letzten Jahren konnte ein stetiger Anstieg des elektrischen Energieverbrauchs in Kraftfahrzeugen beobachtet werden. Dieser Trend wird sich in Zukunft noch verstärken. Zur Reduzierung des Kraftstoffverbrauchs und der Schadstoffemissionen werden weitere Komponenten durch elektrische Komponenten ersetzt (z. B. Servolenkung, elektromagnetische Ventilsteuerung des Motors, elektromechanische Bremsen, usw.). Um den dadurch wachsenden Konsum elektrischer Sekundärenergie befriedigen zu können, präsentierten BMW und DELPHI Automotive Systems im Jahr 2002 ein Brennstoffzellen-Hilfsenergiesystem in einem BMW 750i. Ein 5 kW Hochtemperaturbrennstoffzellensystem (Abk. SOFC; engl.: Solid Oxide Fuel Cell) wandelt die im Diesel enthaltene chemische Energie direkt in elektrische Energie um und erreicht damit einen wesentlich höheren Wirkungsgrad als ein vom Motor angetriebener Generator. Ein Nachteil dieses Hochtemperaturbrennstoffzellensystems ist die Anlaufzeit von einer Stunde [69].

Neben den bereits genannten Vorteilen der niedrigen Schadstoffemission sowie der Unabhängigkeit vom Erdöl ergeben sich aus dem elektrischen Antrieb des Brennstoffzellensystems verbesserte fahrdynamische Eigenschaften. Zudem resultiert aus dem geräusch- und vibrationsarmen Antrieb ein erhöhter Fahrkomfort. Ein weiterer Vorteil des Brennstoffzellenantriebs ist die Verdopplung des Wirkungsgrades im Vergleich zu konventionellen Kraftfahrzeugen mit Verbrennungsmotor [34]. Aus den genannten Gründen sehen die Automobilhersteller langfristig den Einsatz von Brennstoffzellen in ihren strategischen Ausrichtungen vor. Das strategische Vorgehen für die Realisierung dieses Ziels lässt sich in die folgenden Phasen einteilen. Kurzfristig steht die Optimierung der Verbrennungsmotoren im Fokus, anschließend erfolgt die Weiterentwicklung konventioneller Kraftstoffe. Mittelfristig soll der Einsatz von alternativen Kraftstoffen, wie Biodiesel, Erdgas und synthetischen Kraftstoffen, optimiert werden. Durch den Einsatz der sogenannten schwefel- und aromatenfreien Fischer-Tropsch-Kraftstoffe oder auch GTL-Kraftstoffe lassen sich die Partikelemissionen reduzieren. GTL steht hierbei für Gas-to-liquid und für ein Verfahren, das schon in den 20er-Jahren des letzten Jahrhunderts in Deutschland erfunden und in letzter Zeit wiederentdeckt wurde: die Fischer-Tropsch-Synthese, mit der sich aus Erdgas, aber auch aus Kohle und Biomasse verflüssigte Kohlenwasserstoffe herstellen lassen. Der Kraftstoff zeichnet sich durch eine hohe Zündwilligkeit aus und wird von einigen Mineralölkonzernen in geringen Prozentanteilen Dieselmotorkraftstoffen beigemischt [65]. Ein Nachteil dieses synthetischen Kraftstoffs besteht darin, dass es sich um einen Kohlenwasserstoff handelt, d. h., der Wasserstoff ist an Kohlenstoff gebunden. Bei der Verbrennung mit Sauerstoff entsteht demnach weiterhin Kohlendioxid.

Im anschließenden Schritt ist der Umstieg auf Hybridantriebe geplant und parallel dazu der Aufbau einer Wasserstoffinfrastruktur, um so den Weg für die Brennstoffzellentechnologie zu ebnen. Langfristig dienen die Erfahrungen mit den Elektroantrieben der Hybridfahrzeuge im alltäglichen Gebrauch als Basis für den Umstieg auf die Brennstoffzellentechnologie [113].

Für den Einsatz an Bord von Passagierflugzeugen stellen Brennstoffzellensysteme ebenfalls eine emissionsarme, hocheffiziente Alternative zu den konventionellen elektrischen Energieversorgungssystemen dar. Zudem wird über die Nutzung von Wasserstoff als Primärenergieträger nachgedacht. Durch den Austausch des Energieträgers Kerosin mit Wasserstoff ist eine deutliche Reduzierung der Treibhausgasemissionen möglich. Im CRYOPLANE-Forschungsprojekt der EADS gemeinsam mit 35 Verbundpartnern wurden im Zeitraum von 2000 bis 2002 theoretische Flugzeugkonzepte entwickelt, die eine Nutzung von Wasserstoff in den Triebwerken vorsahen. Die Analyse der Konzepte ergab, dass sowohl die Flugzeugleermasse aufgrund des hohen Gewichts der Wasserstofftanks als auch der Energieverbrauch, bedingt durch die erhöhte umströmte Flugzeugfläche, ansteigen. Zudem nehmen die direkten Betriebskosten um ca. 4 % zu. Als Folge dieser Ergebnisse wurde die Weiterentwicklung eines wasserstoffbasierten Flugzeugantriebs nicht weiterverfolgt [49].

Zum gegenwärtigen Zeitpunkt wird von Boeing ein Brennstoffzellendemonstrator-Flugzeug entwickelt. In Abbildung 5-6 ist das Konzept dieses zweisitzigen Flugzeugs dargestellt. Der Verbrennungsmotor einer Dimona HK36TTC wurde durch ein Brennstoffzellensystem ersetzt. Der PEM-Brennstoffzellenstack der Firma Intelligent Energy liefert eine elektrische Leistung von 18 kW, die für den Reiseflug ausreicht. Für den Start sowie für den Steigflug werden zusätzlich Lithium-Ion-Batterien eingesetzt. Der Wasserstoffvorrat ist für

Gesamtkonzeptevaluierung

eine siebenminütige Start-Steigphase und eine anschließend fünfzehnminütige Reiseflugphase in einer Höhe von 1000 Metern ausgelegt. Für die konstante Druckbereitstellung des Kathodengases wird ein Kompressor eingesetzt. Der Erstflug dieses Demonstrators erfolgte Ende 2007 [98].

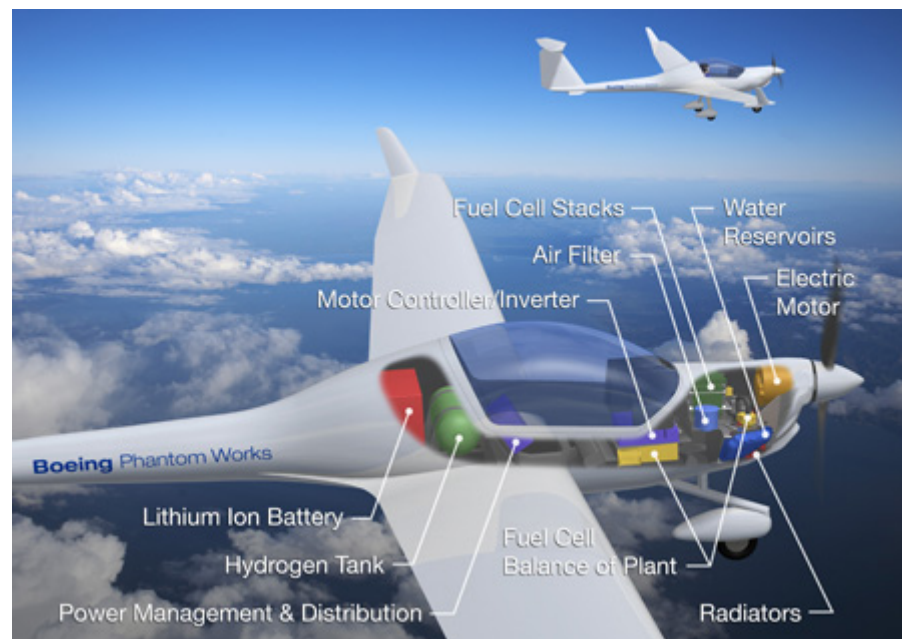


Abbildung 5-6 Brennstoffzellendemonstrator des Flugzeugherstellers Boeing [98]

Wesentlich dringender ist die Bereitstellung von elektrischer Energie an Bord von Flugzeugen für die Energieversorgung. Seit Jahren zeichnet sich ein stetiger Anstieg der an Bord benötigten elektrischen Energie ab. In konventionellen Flugzeugen wird die Energieversorgung im Flugbetrieb von triebwerksgetriebenen Generatoren, Hydraulikpumpen und der aus den Triebwerken entnommenen Zapfluft sichergestellt. Aus Gründen der Effizienzsteigerung und letztlich zur Senkung des Energieaufwands bzw. des Kraftstoffverbrauchs erfolgt eine Umstellung der hydraulischen und pneumatischen Flugzeugsysteme auf elektrische Systeme, wodurch sich der Trend des steigenden elektrischen Energiebedarfs zusätzlich intensiviert [12]. In Abbildung 5-7 ist der Trend der zunehmenden Elektrifizierung an Bord von Passagierflugzeugen dargestellt. Es wird deutlich, dass gerade in den letzten Jahren, bedingt durch die Umstellung auf ein „More Electrical Aircraft“ (Abk. MEA), der Bedarf an elektrischer Energie exponentiell angewachsen ist.

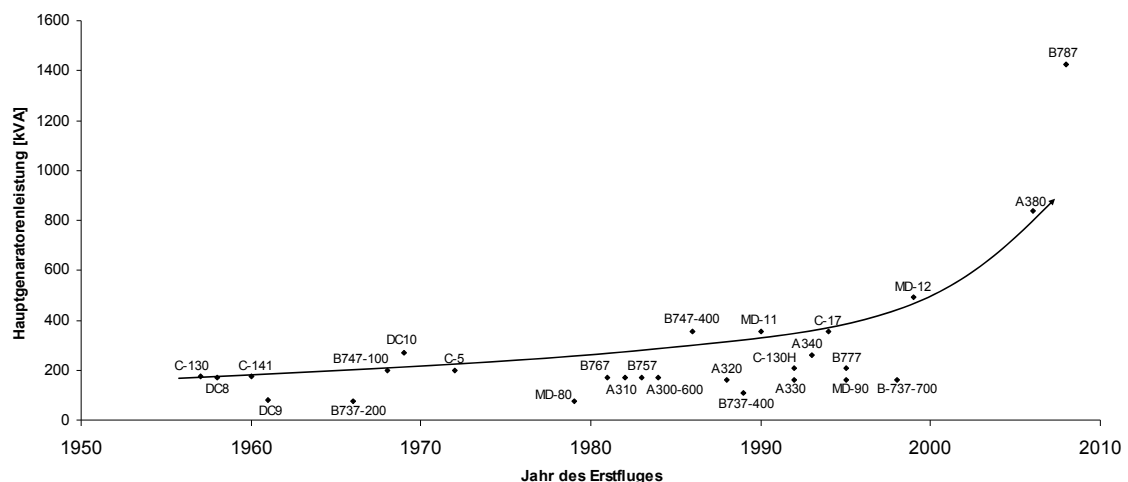


Abbildung 5-7 Anstieg der Triebwerksgeneratorleistung in den vergangenen 40 Jahren [12]

Um den zukünftigen Anforderungen für die Bereitstellung der elektrischen Sekundärenergie gerecht zu werden, bietet die Brennstoffzellentechnologie eine Alternative zu den herkömm-

5.1 Grundlagen von Brennstoffzellensystemen

lichen Energiequellen. Erste Studienergebnisse zeigen ein vielversprechendes Potenzial der Brennstoffzelle im Bereich der Bordnotstromversorgung [43]. Zum gegenwärtigen Zeitpunkt wird die Notstromversorgung im Falle eines Triebwerksausfalls sowie die Versorgung am Boden bei ausgeschalteten Triebwerken von einem Gasturbine-Notstromsystem (Abk. APU; engl.: Auxiliary Power Unit) gewährleistet. Der Betrieb der APU ist mit einer erheblichen Lärm- und Abgasemission verbunden. Die Substitution der APU durch eine Brennstoffzelle bewirkt eine signifikante Reduzierung dieser Emissionen. Stickoxide im Abgas wären nicht vorhanden, und auch der Ausstoß von Treibhausgasen würde sich je nach Brennstoff um 20 % bis 60 % reduzieren lassen. Zudem verspricht die Brennstoffzellentechnologie einen deutlich höheren elektrischen Wirkungsgrad, als ihn die kerosinbetriebene APU mit ca. 15 % aufweist [102]. Für den Einsatz einer Brennstoffzellen-APU sind einige Anforderungen zu erfüllen. Zum einen ist eine elektrische Klimaanlage, zum anderen ein elektrischer Motor zum Start der Triebwerke erforderlich, da die konventionelle APU neben der elektrischen Energie auch pneumatische Energie für den Triebwerksstart sowie die Aufrechterhaltung des Kabinendrucks im Flugbetrieb bereitstellt. In Abbildung 5-8 ist links eine konventionelle APU und rechts eine Brennstoffzellen-APU dargestellt. Bei diesem Brennstoffzellensystem handelt es sich um ein redundantes PEM-Brennstoffzellensystem mit ATR-Reformer.

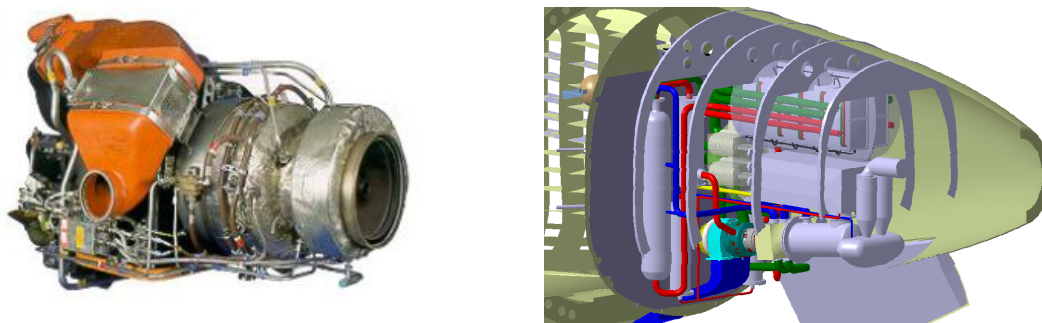


Abbildung 5-8 Konventionelle APU (links) [63] und Brennstoffzellen-APU (rechts) [141]

Zusammenfassend lässt sich sagen, dass der Einsatz von Brennstoffzellen an Bord eines Flugzeuges zur Entlastung der Triebwerksgeneratoren führt. Insgesamt ergibt sich, bedingt durch den höheren Wirkungsgrad gegenüber den Verbrennungsmotoren, eine Reduzierung des Kraftstoffverbrauches. Zusätzlich werden die Geräusch- und Schadstoffemissionen reduziert. Durch die Produktion von Wasser in der Brennstoffzelle verringert sich zudem die Abflugmasse. Nachteile bestehen in der Technologiereife der Brennstoffzellen. So kommen für den kommerziellen Einsatz zum gegenwärtigen Zeitpunkt nur wenige Brennstoffzellentypen, z. B. die PEMFC, infrage. Zudem ist die spezifische Leistung mit durchschnittlich 0,2 kW/kg für PEMFC gering (Mittelwert aus 40 PEM-Brennstoffzellen). Ein weiterer Nachteil besteht in der geringen Lebensdauer der Membran von wenigen 1000 Stunden. Je nach Zelltyp wird die Lebensdauer der Membran zusätzlich durch Verunreinigungen im Anodenbrenngasgemisch, z. B. Schwefel oder Kohlemonoxid, drastisch reduziert. Es ist damit zu rechnen, dass die beschriebenen Probleme in den kommenden Jahren gelöst werden.

Die Vorteile und Potenziale der Brennstoffzellentechnologie haben dazu geführt, dass der Einsatz von Brennstoffzellen an Bord von Flugzeugen in der strategischen Ausrichtung der Flugzeughersteller berücksichtigt wird. In einem ersten Schritt werden zunächst Machbarkeitsanalysen mithilfe von Demonstratoren durchgeführt. Zunächst sind Systeme im niedrigen Leistungsbereich bis ca. 20 kW geplant, für die der benötigte Wasserstoff in Tanks mitgeführt wird. Aufgrund einer fehlenden Wasserstoffinfrastruktur erfolgt anschließend die Entwicklung von Brennstoffzellensystemen, in denen der benötigte Wasserstoff aus dem mitgeführten Kerosin reformiert wird. In dieser Phase werden konventionelle Systeme durch die Brennstoffzellensysteme ersetzt. Mittelfristig sollen Brennstoffzellensysteme für die Primärenergieversorgung eingesetzt werden, um somit auf die Triebwerksgeneratoren verzichten zu können. Langfristig sollen alternative Primärenergieträger, wie z. B. Wasserstoff oder Methanol, zum Einsatz kommen, um so die Entwicklung eines rein elektrischen Flugzeuges zu ermöglichen. In diesen zukünftigen Flugzeugen sollen alle Systeme elektrisch betrieben werden, dies gilt auch für die Triebwerke. Die NASA sieht die Realisierung dieser Flugzeuggeneration allerdings erst gegen Ende des 21. Jahrhunderts [25].

5.2 Realisierung des Simulationskonzepts

In diesem Kapitel wird die Umsetzung des durchgängigen Simulationskonzepts anhand eines Brennstoffzellenforschungsprojekts beschrieben. In chronologischer Reihenfolge wird von der Konzeptphase bis zum Ende der Testphase der kontinuierliche Simulationseinsatz in den einzelnen Schritten des V-Modells dargestellt. Es wird gezeigt, dass sich durch den frühen und durchgängigen Simulationseinsatz ein quantifizierbarer Nutzen des in Kapitel 4 vorgestellten Konzepts für den Entwicklungsprozess von Flugzeugsystemen ergibt.

5.2.1 Architekturmodell des Brennstoffzellensystems

In der frühen Entwicklungsphase (M4–M5; vgl. Abbildung 3-1) ist es zunächst wichtig, verschiedene Konzeptalternativen zu vergleichen und zu bewerten. Bisher basierte die Konzeptauswahl auf empirischem Wissen der an der Entscheidungsfindung beteiligten Mitarbeiter. Durch den Einsatz von Simulationstechniken in der Konzeptphase wird die bisher qualitative Entscheidungsfindung zusätzlich quantifiziert. Einzelne Konzepteigenschaften können direkt anhand von Zahlenwerten evaluiert werden. Dies führt zu einer Optimierung der Entscheidungsfindung und damit zu einer Erhöhung der Erfolgsaussichten. Es wird nicht länger ein geeignetes Konzept für die Weiterentwicklung ausgewählt, sondern das den Anforderungen entsprechend am besten geeignete Konzept.

Für die Integration der Brennstoffzelle an Bord eines Flugzeuges sind verschiedene Konzepte denkbar. Im Folgenden werden sechs Brennstoffzellenkonzepte simuliert und anschließend mit konventionellen Systemen zur Energieversorgung verglichen, um so den Einsatz von Brennstoffzellen auf Machbarkeit hin zu untersuchen.

Für die Simulation der Konzepte wurde eine Brennstoffzellenbibliothek mit dem Simulationsprogramm Matlab/Simulink erstellt. In dieser Bibliothek sind Blöcke der Systemkomponenten enthalten. Die Modularisierung der Komponenten unterstützt den Aufbau verschiedener Konzeptalternativen und ermöglicht durch die Wiederverwendung der Systembausteine eine direkte Vergleichbarkeit. Die Stoffströme zwischen den verschiedenen Komponenten der Modelle sind durch einen vordefinierten Vektor mit einheitlicher Belegung realisiert. Dieser Vektor stellt eine standardisierte Schnittstelle dar und trägt somit zur Erhöhung der Wiederverwendung der Modelle sowie zur Vereinfachung der Modellierung bei. Der Vektor besteht aus 30 Zeilen, in denen die thermodynamischen Eigenschaften, der molare Fluss sowie die chemische Zusammensetzung des Stoffstroms enthalten sind (siehe Anhang 7.7). Bei den in der Bibliothek hinterlegten Modellen handelt es sich um Architekturmodelle (vgl. Abbildung 2-7), d. h., mit diesen Modellen ist eine rein statische Analyse der Konzepte möglich. Die Modellierung basiert auf analytischen Lösungen, womit der Modellierungsaufwand im Gegensatz zu höheren Modelldetaillierungsebenen für die einzelnen Systemkomponenten gering ist. Bedingt durch die geringe Detaillierung ist der Aufbau einer umfangreichen Modellbibliothek, die sich einfach erweitern lässt, im Rahmen dieser Arbeit möglich gewesen. Dieses Vorgehen führt dazu, dass in der frühen Entwicklungsphase eine Großzahl verschiedener Konzepte simuliert und damit quantitativ bewertet werden kann. Das aus der Simulation der Architekturmodelle hervorgehende Konzept wird in weiteren Entwicklungsschritten detailliert. In Analogie zum Top-down-Prinzip im linken Strang des V-Modells steigt mit voranschreitendem Prozess der Detaillierungsgrad der Einzelmodelle an, während gleichzeitig die Anzahl der Modellalternativen sinkt.

In einem ersten Schritt zur Integration von Brennstoffzellen an Bord von Flugzeugen sieht die Strategie der Luftfahrtindustrie (vgl. Kapitel 5.1) zunächst den Einsatz von Systemen im Leistungsbereich von 20 kW mit Sauerstoff- und Wasserstoffdruckspeichern vor. Aus diesem Grund wurden zwei Konzepte mit einer elektrischen Leistung von 20 kW und Druckspeichern modelliert. Die Konzepte unterscheiden sich durch die Bereitstellung des Sauerstoffs. In Konzept 1 wird dieser mittels einer Druckflasche bereitgestellt. Im zweiten Konzept wird die Brennstoffzelle mit Luft betrieben; dieses Konzept erfordert in Abhängigkeit zur Flughöhe den Einsatz eines Kompressors. In beiden Konzepten wird die Abwärme einer PEM-Brennstoffzelle über einen Kreuzstromwärmetauscher an die Umgebungsluft abgegeben. Die Konzepte werden als mögliche Alternativen zur derzeitigen Staudruckturbine (Abk. RAT; engl.: Ram Air Turbine) angesehen. Bei diesem System handelt es sich um eine zusätzliche Notfallenergieversorgung, die im Falle eines Ausfalls sämtlicher Triebwerksgeneratoren und/oder Hydraulikpumpen eingesetzt wird. In Tabelle 5-3 sind die Ergebnisse der Simulation sowie die Werte einer A320-RAT von Hamilton Sundstrand dargestellt [62]. Die Umgebungs-

5.2 Realisierung des Simulationskonzepts

bedingungen für die Simulation entsprechen dem internationalen Standardatmosphärenmodell in einer Höhe von 10 km.

		Konzept 1 PEMFC	Konzept 2 PEMFC/Kompressor	Staudruckturbine ERPS08
Gewicht	kg	117	153	83
Leistung	kW	18,7 elektrisch	18,7 elektrisch	18,67 hydraulisch
spezifische Leistung	kW/kg	0,16	0,12	0,23
Systemwirkungsgrad	-	45 %	48 %	k. a.
Eingangsgroößen		0,28 g/s H ₂ 2,21 g/s O ₂	0,28 g/s H ₂ 19,12 g/s Luft	Min. Anströmgeschwindigkeit 48 m/s

Tabelle 5-3 Vergleich der Simulationsergebnisse mit einer konventionellen RAT

Um den Einfluss des Brennstoffzellenarbeitspunkts in der Simulation zu berücksichtigen, wurde das Gewicht der Brennstoffzelle auf die Membranfläche bezogen (mittleres flächenbezogenes Gewicht aus 40 PEM-Brennstoffzellen). Setzt man eine konstante Leistung der Brennstoffzelle voraus, ergeben sich im Bereich hoher Spannungen ebenfalls hohe Massen. Wird der Arbeitspunkt zu niedrigeren Spannungen verschoben, steigt die Stromdichte, d. h., die Membranfläche wird bei konstanter Leistung kleiner (vgl. Abbildung 5-2). Dies hat zur Folge, dass auch die Masse der Brennstoffzelle sinkt. Eine höhere Stromdichte führt zu einer Zunahme der Abwärme und damit zu einer Reduzierung des Wirkungsgrades. In beiden Konzepten wurde der Arbeitspunkt bei einer Zellspannung von 0,7 V und einer Leistung von 18,7 kW festgelegt. Hieraus ergibt sich mit Gleichung (5-4) ein konstanter Strom und damit ein konstanter molarer Wasserstofffluss (siehe Tabelle 5-3) [99].

$$I = 2 \cdot F \cdot \dot{n} \quad (5-4)$$

Dabei bezeichnet I den Strom, F die Faraday-Konstante mit 96.485 Coulomb und die erste Ableitung von n nach der Zeit den molaren Wasserstoffstrom. Bedingt durch den Anstieg der Leerlaufspannung, hervorgerufen durch den höheren Sauerstoffpartialdruck, verschiebt sich der Arbeitspunkt des ersten Konzepts bei konstanter Leistung zu höheren Stromdichten i. Dieser Zusammenhang ist in Abbildung 5-9 dargestellt.

Da der Strom ebenfalls konstant ist und sich die Membranfläche A umgekehrt proportional zur Stromdichte verhält, sinkt die Fläche mit steigender Stromdichte. Dieser Zusammenhang ist in Gleichung (5-5) dargestellt:

$$i = \frac{I}{A} \quad (5-5)$$

Der Betrieb einer Brennstoffzelle mit höherem Sauerstoffpartialdruck führt bei den beschriebenen Eingangsparametern zu einer Verkleinerung der Membranfläche. Im Konzept 1 beträgt die Fläche 3,3 m², während sie für Konzept 2 6,5 m² beträgt. Da sich die Masse der Brennstoffzelle proportional zur Fläche verhält, ergibt sich für das Brennstoffzellengewicht des zweiten Konzepts eine Masse von 132 kg und damit ein Gewichtsanteil von 86 % der Gesamtsystemmasse. Bedingt durch die höhere Stromdichte des ersten Konzepts ergibt sich bei konstanter Leistung eine höhere Verlustleistung der Brennstoffzelle (vgl. Abbildung 5-2). Obwohl im zweiten Konzept der Kompressor als zusätzlicher Verbraucher den Nenner der Wirkungsgradberechnung erhöht, wirkt sich die erhöhte Stromdichte stärker auf den Wirkungsgrad aus, sodass der Wirkungsgrad des ersten Konzepts unter dem des zweiten liegt.

In Tabelle 5-3 wurde deutlich, dass die spezifischen Leistungen der Konzepte unterhalb der Leistung einer konventionellen Staudruckturbine liegen. Die RAT stellt im Gegensatz zu den Brennstoffzellenkonzepten eine hydraulische Leistung zur Verfügung. Mithilfe eines Hydraulikmotors, der einen Generator antreibt, kann die RAT eine elektrische Leistung von 5 kW zur Verfügung stellen. Im Falle des Ausfalles sämtlicher Triebwerksgeneratoren kann ein Brennstoffzellensystem ohne vorherige Energiewandlung elektrische Leistung bereit-

stellen, während im Falle des Ausfalls sämtlicher Triebwerkshydraulikpumpen eine Umwandlung in der RAT unnötig ist.

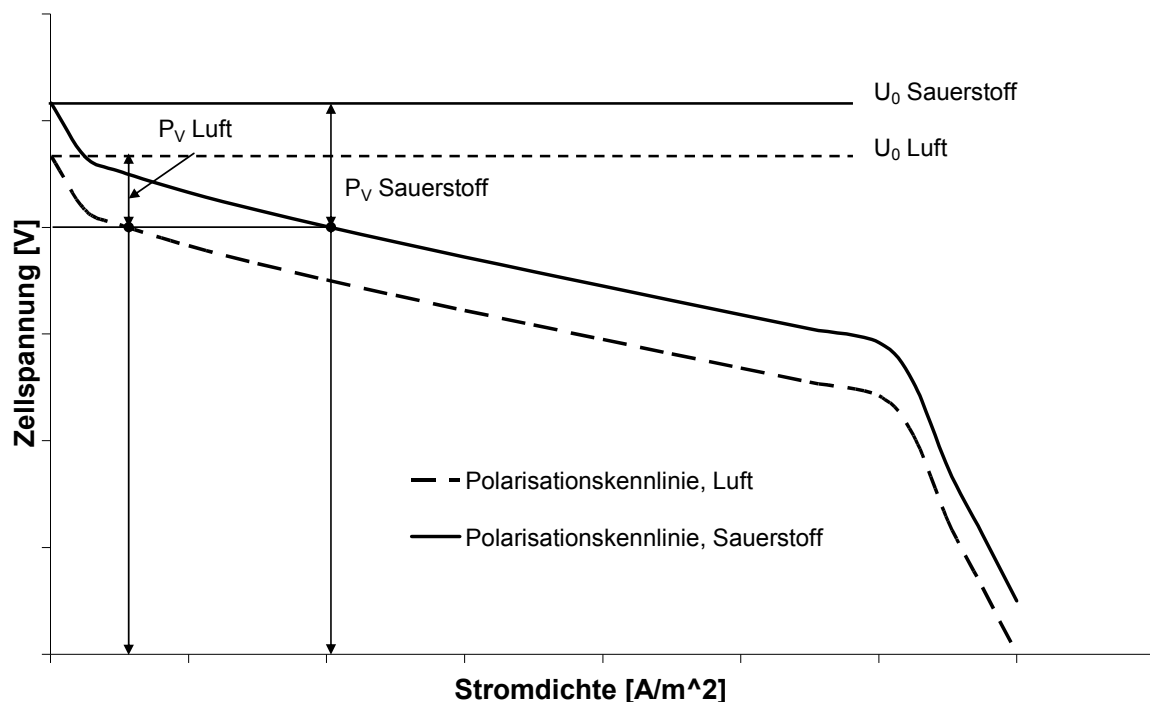


Abbildung 5-9 Verschiebung der Stromdichte bei konstanter Spannung und Leistung

Die Staudruckturbine benötigt eine minimale Anströmgeschwindigkeit von 48 m/s. Im Notfall bedeutet dies, dass die RAT die Notenergieversorgung bis zur Landung bereitstellt. Der Druckflascheninhalt der Brennstoffzellenkonzepte muss hingegen für eine bestimmte Dauer ausgelegt werden; in der Simulation betrug die Auslegungsdauer des Brenngas- und Sauerstoffvorrats eine Stunde. Da der genaue Tankvorrat nicht bestimmt werden kann und der Notfall nur in den seltensten Fällen eintritt, bedeuten die Gasflaschen sowie deren Inhalt zusätzlichen Ballast und führen zur Erhöhung der direkten Betriebskosten des Flugzeugs (Abk. DOC; engl.: Direct Operating Costs). An dieser Stelle sei darauf hingewiesen, dass nur das reine Systemgewicht betrachtet wurde. Eventuelle Modifikationen an der Flugzeugstruktur wurden nicht berücksichtigt. Zusätzlich führt die Installation der RAT im Flügel zu einer permanenten Erhöhung des Luftwiderstands, der sich wiederum negativ auf den Treibstoffverbrauch auswirkt.

Mit den derzeit geringsten PEM-Brennstoffzellengewichten von $9,75 \text{ kg/m}^2$ kann die Masse des Stacks halbiert werden. In diesem Fall erreichen die beiden Konzepte eine spezifische Leistung von ca. $0,21 \text{ kW/kg}$, dieser Wert entspricht in etwa dem der RAT. Aus Sicht der Simulationsergebnisse ist der Einsatz einer Brennstoffzellennotenergieversorgung durchaus denkbar. Für den Einsatz an Bord gilt es, die in der Simulation nicht berücksichtigten Eigenschaften, wie z. B. besondere Sicherheitsvorschriften im Umgang mit Wasserstoff, aber auch die Reduzierung des Luftwiderstands, gegeneinander abzuwägen.

Im folgenden Schritt der Brennstoffzellenstrategie sieht die Luftfahrtindustrie den kontinuierlichen Einsatz der Brennstoffzelle an Bord vor (vgl. Kapitel 5.1). Bedingt durch das Fehlen einer geeigneten Wasserstoffinfrastruktur soll der benötigte Wasserstoff aus dem mitgeführten Kerosin reformiert werden. Durch den Einsatz eines Brennstoffzellensystems auf Basis der Kerosinreformierung ist eine autonome elektrische Energieversorgung am Boden möglich, sodass die kostenintensive elektrische Energieversorgung an den Flughäfen überflüssig wird. Um die Flughafengebühren weiter zu reduzieren, wird über eine Verringerung der Emissionen nachgedacht. Die Fortbewegung des Flugzeugs am Boden wird derzeit von den Haupttriebwerken übernommen. Ihr Betrieb ist mit einer hohen Geräusch- und Abgasemission verbunden, zusätzlich ist mit diesem Antrieb ein Rückwärtsfahren nicht möglich. Ein elektrischer Radantrieb würde diese drei Probleme lösen. Die Bereitstellung der elektrischen Energie kann ebenfalls durch den Einsatz eines Brennstoffzellensystems erfolgen. In Kapitel 5.1 wurde beschrieben, dass die PDh die derzeit einzige emissionsfreie

5.2 Realisierung des Simulationskonzepts

Reformierungsmethode darstellt. Durch den Einsatz der PDh ist damit theoretisch die emissionsfreie Fortbewegung des Flugzeugs am Boden möglich (engl.: zero emission taxiing). Aus diesem Grund werden im Folgendem zwei Konzepte mit partieller Dehydrogenierung zur kontinuierlichen elektrischen Energieversorgung untersucht. In Abbildung 5-10 ist der Systemaufbau der Konzepte 3 und 4 dargestellt. Wie bei den Konzepten 1 und 2 wird eine elektrische Leistung von 20 kW bereitgestellt. Zusätzlich zur Stromversorgung sollen mit dem Brennstoffzellenabgas die Treibstofftanks des Flugzeugs inertisiert werden. Um die Selbstentzündung des Gasgemisches in den Flugzeugtanks zuverlässig zu vermeiden, ist ein Sauerstoffgehalt von unter 12 % erforderlich [169]. Derzeit werden in den Flugzeugtanks beschichtete Kohlenstofffasern eingesetzt. In Luftfiltermodulen (Abk. ASM; engl.: Air Separation Module) wird der Luftsauerstoff in einem katalytischen Prozess abgespalten, zusätzlich sind die Fasern im geringen Maße sauerstoffdurchlässig [126]. Nachteile dieser AS-Module sind die hohen Anschaffungs- und Wartungskosten. Zudem wird die Lebensdauer der Fasern durch den Kontakt mit Ozon stark reduziert. Durch den Einsatz der Simulation soll geprüft werden, ob sich das Brennstoffzellenabgas zur Tankinertisierung eignet und damit den Einsatz der ASM überflüssig macht.

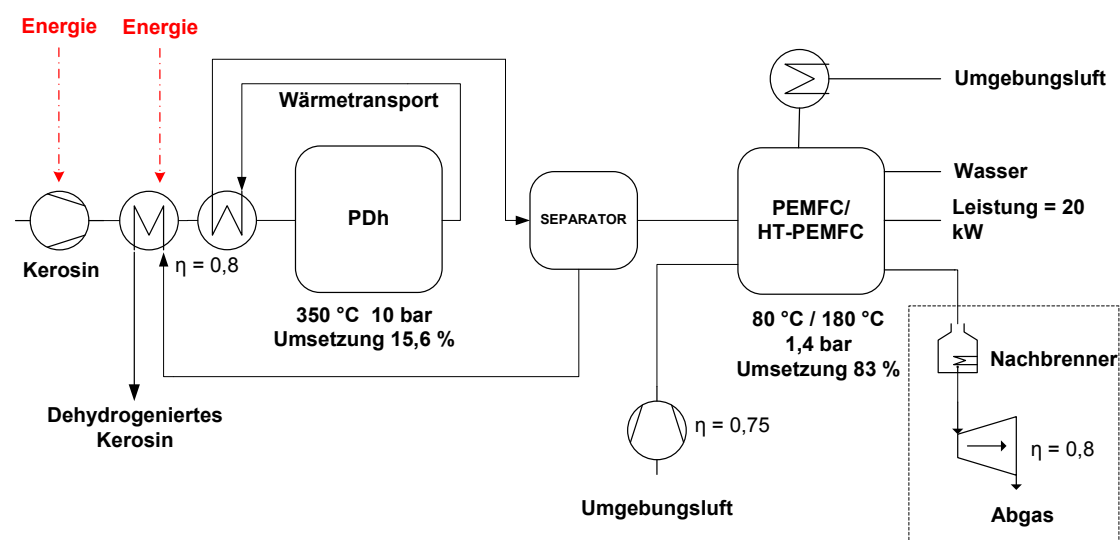


Abbildung 5-10 Brennstoffzellensystemkonzepte mit partieller Dehydrogenierung

Die Konzepte 3 und 4 unterscheiden sich lediglich durch die Verwendung einer Hochtemperaturbrennstoffzelle bei einer Betriebstemperatur von 180 °C bzw. Niedertemperaturbrennstoffzelle bei einer Betriebstemperatur von 80 °C. Durch den Einsatz einer Hochtemperaturbrennstoffzelle vereinfacht sich zum einen das Wassermanagement der Zelle, zum anderen kann der Wärmestrom der Brennstoffzelle über eine größere Temperaturdifferenz an die Umgebung abgeführt werden, wodurch sich die Austauschfläche und damit das Gewicht des Wärmetauschers reduziert. In einer Studie der Volkswagen AG wurde gezeigt, dass bei gleicher Radleistung bei einem Brennstoffzellensystem mehr als die doppelte Wärmemenge im Gegensatz zum Verbrennungsmotor über das Kühlmittel abgeführt werden muss. Dies hängt damit zusammen, dass ein Großteil der Abwärme bei den Verbrennungsmotoren über das heiße Abgas abgeführt wird. Unter Verwendung der gleichen Kühlungscharakteristik des Fahrzeuges wird deutlich, dass unter Extrembedingungen (z. B. Steigfahrt oder Heißtag) bei einer Temperaturdifferenz von 40 °C, keine vergleichbaren Fahrleistungen wie mit den konventionellen Antrieben erreicht werden, da die entsprechende im Brennstoffzellensystem anfallende Wärmemenge nicht an die Umgebung abgeführt werden kann. Die Volkswagen AG zeigt in der Studie, dass es mit einer Hochtemperatur-PEM-Brennstoffzelle möglich ist, annähernd die Fahrleistungen eines vergleichbaren konventionellen Fahrzeuges zu erreichen [158].

In den beiden Konzepten zur kontinuierlichen Energieversorgung wird die Brennstoffzelle mit Umgebungsluft betrieben, die durch einen Kompressor auf den Systembetriebsdruck gebracht wird. Das Kerosin wird einem Flugzeugtank entnommen und in einem Vorwärmprozess auf die Betriebstemperatur des PDh-Reaktors erwärmt. Nach der chemischen Reaktion werden die Produkte gekühlt und an einen Separator in Wasserstoff (gasförmig), Kerosin (flüssig) und dehydriertes Kerosin (flüssig) aufgeteilt. Die entzogene Wärmemenge

Gesamtkonzeptevaluierung

wird zum Vorheizen des Kerosinstroms benutzt. Das dehydrierte Kerosin sowie das nicht umgewandelte Kerosin werden abgekühlt und stehen zur Verbrennung in den Triebwerken zur Verfügung. Der entnommene Wärmestrom wird ebenfalls zur Kerosinvorheizung verwendet. Die Abwärme der Brennstoffzelle wird mittels eines Rohr/Platten-Kreuzstrom-wärmeübertragers an die Umgebungsluft abgegeben. Um ausreichende Fluidströme für den Wärmeaustausch zu realisieren, sind ein Gebläse für die Kühlluft und eine Pumpe für das Kühlwasser vorhanden. Der nicht umgesetzte Wasserstoff wird in einer Brennkammer verbrannt und über eine Turbine expandiert. In Abbildung 5-11 ist ein Ausschnitt des Matlab/Simulink-Modells des dritten Konzepts dargestellt.

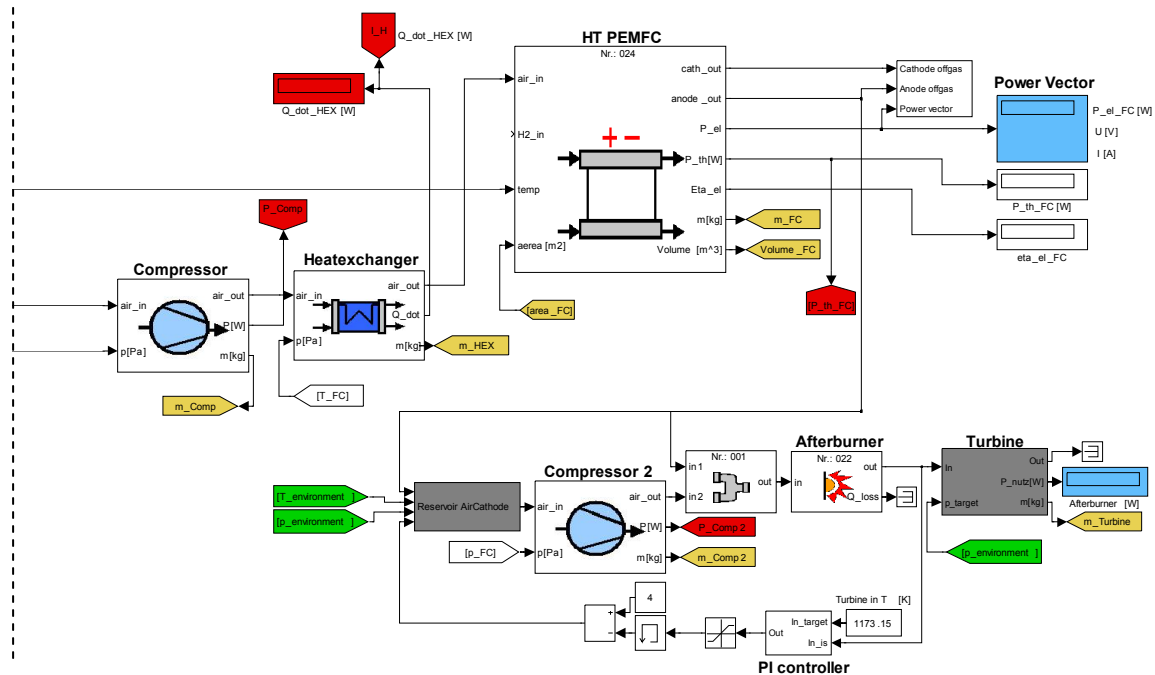


Abbildung 5-11 Ausschnitt: Simulationsmodell Konzeptalternative 3

Im Flugzeugbau wirkt sich jede zusätzliche an Bord installierte Masse auf die direkten Betriebskosten aus. Aus diesem Grund ist eine Gewichtsoptimierung des Brennstoffzellensystems von besonderer Bedeutung. In einer Design of Experiments-Studie (Abk. DoE) wurde der in Bezug auf die Systemmasse optimale Betriebspunkt des Brennstoffzellensystems ermittelt. Hierfür wurde die Zellspannung im Bereich von 0,9 bis 0,5 V schrittweise variiert. Dieser Bereich entspricht dem ohmschen Teil der Polarisationskurve der Brennstoffzelle. Der Leerlaufbereich und der Kurzschlussbereich wurden nicht in die Analysen einbezogen. Aufgrund von Aktivierungsverlusten im Leerlaufbereich und Massentransportverlusten im Kurzschlussbereich weist die Kennlinie der Brennstoffzelle hohe Spannungsgradienten auf. Diese machen eine Regelung der Zelle in diesem Bereich sehr schwierig. Zudem bewirkt der Betrieb im Leistungsmaximum der Zelle eine erhöhte Degradation der Membran und damit eine Verkürzung der Lebensdauer. Die Sensitivitätsanalysen wurden für vier Umgebungszenarien durchgeführt (siehe Tabelle 5-4).

Szenario	Flughöhe [ft]	Lufttemperatur [°C]	Luftdruck [Pa]	Kerosintemperatur [°C]
ISA Boden	0	15	1,013 e5	15
Heißer Boden	0	56	1,088 e5	56
Kalter Flug	43000	-72	1,124 e4	-33
Kabine Flug	43000	20	7,52 e4	-33

Tabelle 5-4 Umgebungsparameter der Simulationsstudie

So ist es möglich, für jede Systemkomponente das Gewicht im optimalen Betriebspunkt zu bestimmen. Im heißen Bodenbetrieb wird z. B. der Brennstoffzellenwärmetauscher eine höhere Masse aufweisen als unter kalten Flugbedingungen, der Auslegungspunkt für den Wärmetauscher ist somit der heiße Bodenfall. In Abbildung 5-12 ist für das Konzept 3 das Ergebnis der Sensitivitätsanalyse für die Umgebungsbedingung „Kalter Flug“ dargestellt.

5.2 Realisierung des Simulationskonzepts

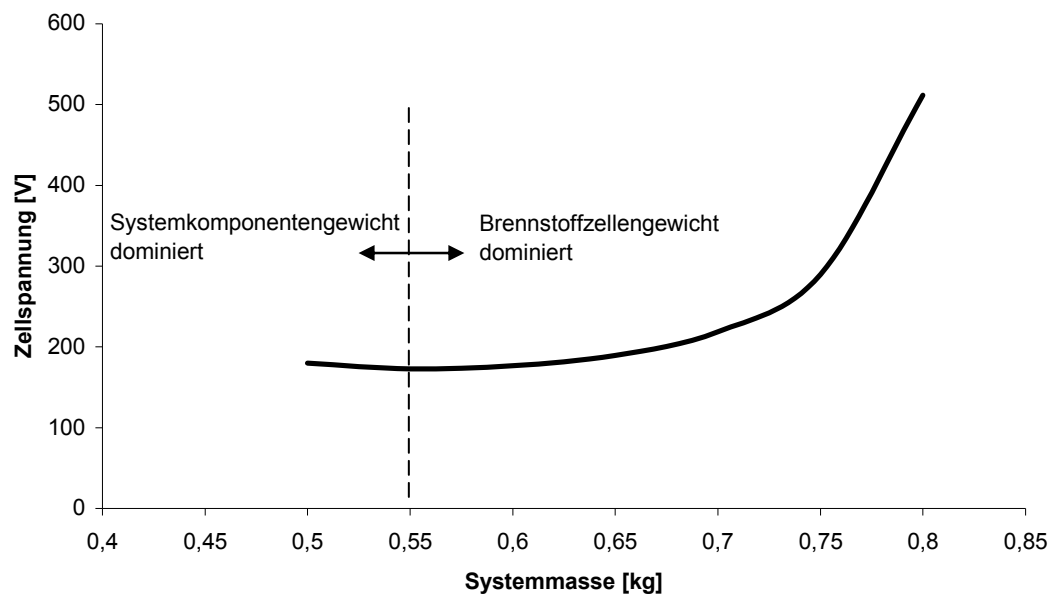


Abbildung 5-12 Sensitivitätsanalyse Konzept 3 „Kalter Flug“

Im Bereich kleiner Spannungen dominieren die Peripheriekomponenten (z. B. Wärmetauscher, Vorheizter, Kompressor usw.) die Systemmasse. In diesem Bereich wird die Brennstoffzelle bei hohen Stromdichten betrieben, d. h., die Brennstoffzelle produziert mehr Abwärme als im Bereich höherer Spannungen, wodurch sich der Brennstoffzellenwirkungsgrad verschlechtert. Mit steigender Spannung wächst der Einfluss des Brennstoffzellengewichts stärker an und dominiert das Systemgewicht. Dieser Effekt ist darin begründet, dass die Stromdichte mit steigender Zellspannung abnimmt. Um die konstante Leistung von 20 kW bereitzustellen, ist somit eine größere Membranfläche erforderlich [vgl. Gleichung (5-5)]. Da sich das Gewicht der Brennstoffzelle proportional zur Membranfläche verhält, steigt das Gewicht der Brennstoffzelle ebenfalls an. Der optimale Betriebspunkt der Brennstoffzelle für Konzept 3 liegt bei 0,55 V. Für diesen Punkt ergibt sich ein minimales Systemgewicht von 173 kg. Wie bereits beschrieben, müssen für die verschiedenen Szenarien die jeweils maximalen Gewichte der Systemkomponenten für den optimalen Betriebspunkt ermittelt werden. Die Anpassung der Gesamtmasse führt wiederum zu einer Änderung des optimalen Betriebspunkts. In Tabelle 5-5 sind die korrigierten Systemmassen und Betriebspunkte für die zwei Konzepte enthalten.

	Konzept 3	Konzept 4	Einheit
Leistung	20	20	kW
Zellspannung	0,6	0,5	V
Systemmasse	182	183	kg
Spezifische Leistung	0,1	0,1	kW/kg
Treibstoff	1,3288	1,4496	mol/s
	748	815	kg/h
Brennstoffzellenwirkungsgrad	48	39	%
Wasserstoff	0,2072	0,2487	mol/s
	1,5	1,8	kg/h
Sauerstoff/Luft	0,9890	1,1868	mol/s
	102	123	kg/h

Tabelle 5-5 Systemmassen und Betriebspunkte der Konzepte 3 und 4

In den folgenden zwei Abbildungen sind die Einzelgewichte der Komponenten dargestellt.

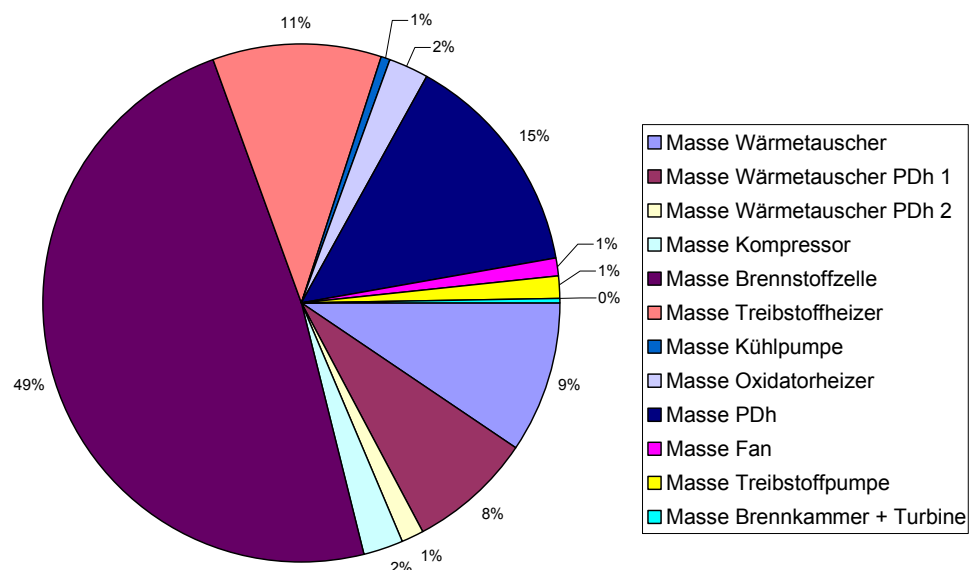


Abbildung 5-13 Komponentenmassen Konzept 3 (mit PEMFC)

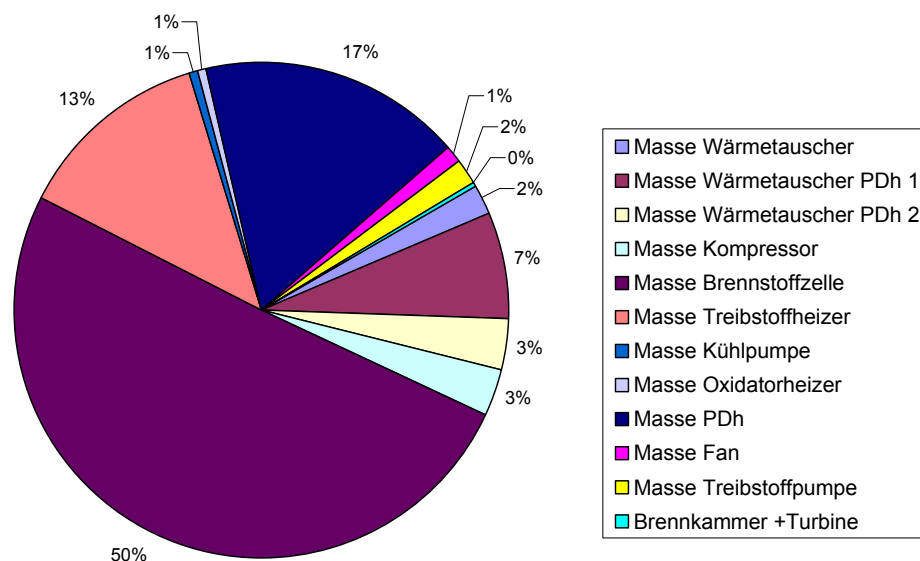


Abbildung 5-14 Komponentenmassen Konzept 4 (mit HT-PEMFC)

In beiden Konzepten überwiegt der Gewichtsanteil der Brennstoffzelle mit ca. 50 %. Den nächst größten Anteil am Gesamtgewicht hat das PDh-System. Der PDh-Katalysator, die Treibstoffpumpe, die PDh-Wärmetauscher sowie der Treibstoffheizer tragen mit 35 % bzw. 40 % zur Systemmasse bei. Der Brennstoffzellenwärmetauscher hat in Konzept 3 einen Anteil von 9 % am Systemgewicht, während er im vierten Konzept nur 2 % des Systemgewichts ausmacht. Dieses Ergebnis stimmt mit der Studie der Volkswagen AG überein [158]. Eine erhöhte Brennstoffzellenbetriebstemperatur führt dazu, dass die Abwärme über eine größere Temperaturdifferenz an die Umgebung abgegeben werden kann. Dies hat zur Folge, dass die Wärmeaustauschfläche und damit das Gewicht des Wärmetauschers des HT-PEMFC-Konzepts kleiner ist als die des PEMFC-Konzepts. Trotz des geringeren Wärmetauschergewichts ist das Gesamtgewicht des HT-PEMFC-Konzepts größer. Der Grund hierfür liegt in dem niedrigeren Brennstoffzellenwirkungsgrad der HT-PEMFC (vgl. Tabelle 5-5). Die

5.2 Realisierung des Simulationskonzepts

Ursache dafür ist, dass die Kennlinie der HT-PEM-Zelle gegenüber der PEM-Zelle zu niedrigeren Spannungen verschoben ist. Bei identischer Stromdichte und Membranfläche liefert die HT-PEM-Zelle also weniger elektrische Leistung. Zusätzlich liegt der Betriebspunkt des vierten Konzepts bei einer kleineren Spannung als im dritten Konzept, wodurch sich der Brennstoffzellenwirkungsgrad ebenfalls verringert. Für die Bereitstellung einer konstanten Leistung von 20 kW wird somit ein höherer Wasserstoff- und damit auch ein höherer Kerosinstrom benötigt. In den Peripheriekomponenten des Systems muss dementsprechend ein größerer Fluidstrom als in Konzept 3 vorgeheizt und befördert werden.

Zur Berechnung des Systemwirkungsgrads wird zunächst die Leistung ermittelt, die dem System zugeführt wird. Sie besteht zum einen aus der Differenz der Heizwertströme $\dot{n} \cdot H_U$ des zugeführten und ablaufenden Treibstoffs, wobei \dot{n} der molare Treibstoffstrom und H_U der untere Heizwert des Treibstoffs ist, zum anderen besteht sie aus sämtlichen elektrischen Leistungen P_{PER} , die von den verschiedenen Systemkomponenten benötigt werden. Der Wirkungsgrad errechnet sich als Quotient der Summe aus elektrischer Leistung der Brennstoffzelle P_{EL} und des Nachbrenners P_A sowie der dem System zugeführten Leistung.

$$\eta_{SYS} = \frac{P_{EL} + P_A}{\dot{n}_{IN} \cdot H_{U_{IN}} - \dot{n}_{OUT} \cdot H_{U_{OUT}} + \sum P_{PER}} \quad (5-6)$$

Vergleicht man die beiden Konzepte an den Betriebspunkten, wird deutlich, dass sich die Systemwirkungsgrade nur marginal unterscheiden. Für die Konzepte 3 und 4 ergibt sich ein Wirkungsgrad von 25 % bzw. 23 %. Der niedrigere Wirkungsgrad des vierten Konzepts ist auf den geringeren Brennstoffzellenwirkungsgrad zurückzuführen.

Die Tankinertisierung hat das Ziel, mithilfe des Kathodenabgases der Brennstoffzelle den Sauerstoffvolumenanteil im Tank unter 12 % zu halten, um so die Selbstentzündung des Gasgemischs zu vermeiden. Die Inertisierungssimulationen für die Konfiguration 4 lieferten folgende Ergebnisse.

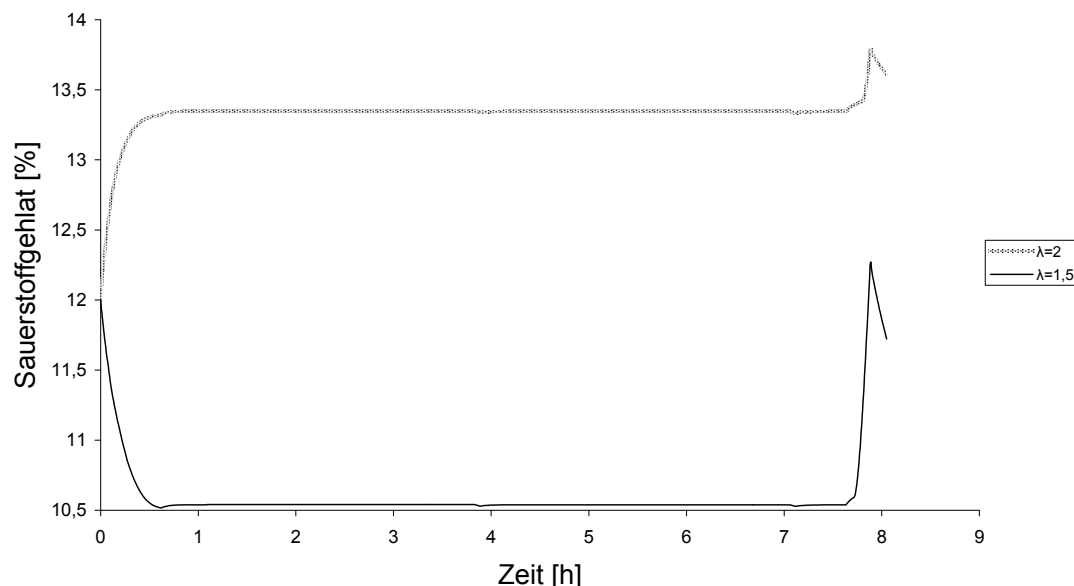


Abbildung 5-15 Sauerstoffgehalt im Flugzeugtank als Funktion der Zeit

Der Sauerstoffanteil des Kathodenabgases liegt für eine Luftüberschusszahl von 2 und eine 83-prozentige Wasserstoffumsetzung in der Brennstoffzelle bei 11,2 %. Nach der Wasserabscheidung erhöht sich der Volumenanteil des Sauerstoffs auf 13,3 % und liegt damit deutlich über dem für die Zertifizierung erlaubten Wert von 12 % [169]. Durch den Einsatz der Simulation wurde damit frühzeitig gezeigt, dass mit den verwendeten Systemparametern eine Gasinertisierung nicht möglich ist. In Abbildung 5-15 wird deutlich, dass durch eine Reduzierung des Luftüberschusses von 2 auf 1,5 die Tankinertisierung prinzipiell möglich ist.

Als Eingangsvariablen der Inertisierungssimulation dienen ein Flugprofil und der Flussvektor des Kathodenausgangs. In dem Flugprofil ist ein achtstündiger Flug hinterlegt, der die Höhe als Funktion der Flugzeit beinhaltet. Aus diesen Eingangsvariablen und den Tankparametern (Volumen, Sauerstoffgehalt zu Beginn des Flugs, Füllstand, Kerosindichte) wird der Sauerstoffgehalt im Tank als Funktion der Flugzeit berechnet. In Abbildung 5-15 ist in der Steigphase von $t = 0$ bis $t = 30$ min für eine Luftüberschusszahl von 1,5 ein Rückgang der Sauerstoffkonzentration erkennbar. Während des Aufstiegs strömt aufgrund des Druckausgleichs zwischen Tank und Umgebung Tankleerraumgas aus den Flugzeugtanks. Gleichzeitig wird sauerstoffarmes Gas aus dem Brennstoffzellensystem in den Tank geleitet, sodass der Sauerstoffgehalt in dieser Phase stark abnimmt. Während des Reiseflugs bleibt der Sauerstoffgehalt konstant. Bedingt durch den Kerosinverbrauch, der nicht vollständig vom Inertisierungsgas kompensiert werden kann, stellt sich im Tank ein Sauerstoffgehalt oberhalb des Sauerstoffgehalts des Inertisierungsgases ein. Während des Reisefluges nimmt die Flugzeugmasse kontinuierlich ab. Um die Fluggeschwindigkeit sowie den Auftriebsbeiwert des Flugzeugs in dieser Flugphase konstant zu halten, wird in der Praxis eine treppenförmige Flugbahn beschrieben. Diese Treppenstufen machen sich in einer Abnahme der Dichte bemerkbar und sind in der Inertisierungskurve erkennbar. Nach diesen Dichteänderungen stellt sich in den Tanks wieder der konstante Sauerstoffgehalt von 10,6 % ein. Während des Sinkflugs von $t = 7,6$ bis $t = 8$ Stunden nimmt der Sauerstoffgehalt stark zu. In dieser Phase strömt Umgebungsluft mit einem Sauerstoffanteil von ca. 21 % in den Tank. Der Sauerstoffgehalt steigt kurzzeitig auf einen Wert von 12,3 %. Nach der Landung sinkt der Sauerstoffgehalt, bedingt durch die kontinuierliche Inertisierung, wiederum schnell ab.

Zusammenfassend lässt sich für die PDh-Konzepte sagen, dass die Systemwirkungsgrade mit 25 % bzw. 23 % unterhalb des Wirkungsgrads moderner Wärmekraftmaschinen, die durch den Carnot-Wirkungsgrad begrenzt sind, liegen. Eine Ursache hierfür ist der Betriebspunkt der Brennstoffzellen bei kleinen Spannungen. Mit steigendem Wirkungsgrad steigen auch die Membranflächen und damit die Masse der Brennstoffzelle an. Zudem benötigen die Peripheriekomponenten Energie, für das Konzept 3 wird unter kalten Flugbedingungen eine Leistung von 60 kW benötigt, dabei entfallen alleine 26 kW auf die endotherme PDh-Reaktion. Einen weiteren Einfluss auf den Systemwirkungsgrad hat die Wärmerückgewinnung. Für die Bereitstellung des Wasserstoffs wird eine große Menge an Kerosin benötigt, wobei 84,4 % dieses Kerosins nicht in der PDh umgesetzt werden [siehe Gleichung (5-3)]. In Konzept 3 werden 748 kg Kerosin pro Stunde benötigt. Dieser Kerosinstrom wird zunächst auf eine Temperatur von 350 °C erwärmt und nach der PDh-Reaktion in einem Wärmetauscher zum Vorwärmen des kalten Kerosinstroms verwendet. Die Effizienz der Wärmetauscher hat damit einen starken Einfluss auf den Systemwirkungsgrad. In der Simulation wurde die Wärmerückgewinnung mit 85 % angenommen, eine Erhöhung dieses Wertes würde sich positiv auf den Systemwirkungsgrad und negativ auf das Systemgewicht auswirken. Die PDh-Reformierung befindet sich zurzeit im Laborstadium. In Zukunft wird die Umsetzungsrate von derzeit 15,6 % durch den Einsatz optimierter Katalysatoren und Reaktorgeometrien gesteigert werden. Hieraus resultiert ein reduzierter Kerosinstrom, der zu einer Verkleinerung der Peripheriekomponenten bei gleichzeitiger Erhöhung des Systemwirkungsgrades führt.

Eine Versorgung der Peripheriekomponenten mit elektrischer Leistung würde wenig sinnvoll sein, da die benötigte Leistung dreimal höher ist als die von der Brennstoffzelle produzierte elektrische Leistung. Aus diesem Grund sollten die benötigten Wärmeströme für die endotherme PDh-Reaktion sowie das Vorwärmen des Kerosins mittels chemischer Verbrennung bereitgestellt werden. Ein emissionsfreier Radantrieb des Flugzeugs ist mit dieser Konfiguration allerdings nicht mehr möglich, da bei der chemischen Verbrennung wiederum Kohlenmonoxid anfällt. Des Weiteren fällt bei der Verbrennung des dehydrierten Kerosins in den Triebwerken bei gleicher Leistung im Vergleich zum normalen Kerosin eine erhöhte Kohlendioxidemission an, da pro dehydriertem Kerosinmolekül ein Wasserstoffmolekül weniger vorhanden ist und somit der Heizwert reduziert ist. Damit können Emissionen zwar lokal reduziert werden, global gesehen wird allerdings die gleiche Menge an Schadstoffen emittiert.

Ein Sicherheitsproblem stellt die große Menge Kerosin bei einer Temperatur von 350 °C dar. Bei Standardumgebungsbedingungen liegt der Siedepunkt für Kerosin je nach Sorte zwischen 175 und 288 °C. Bei einem Leck im System würde damit eine große Menge Kerosingas freigesetzt werden, das in Verbindung mit Luft ein explosives Gasgemisch bildet.

5.2 Realisierung des Simulationskonzepts

Neben den beschriebenen Nachteilen, zu denen auch die geringe spezifische Leistung von 0,1 kW/kg zählt, weisen die PDh-Konzepte auch Vorteile auf. Der kontinuierliche Einsatz der Brennstoffzelle ermöglicht zum einen die Tankinertisierung, lediglich im Sinkflug steigt die Sauerstoffkonzentration für kurze Zeit über den Wert von 12 %. Das von der Brennstoffzelle produzierte Wasser kann zum anderen für die Grauwasserversorgung an Bord eingesetzt werden, womit sich die Abflugmasse des Flugzeugs reduzieren lässt. Eine weitere Anwendung stellt die Injektion des Wassers in die Triebwerken zur Erhöhung des spezifischen Impulses und damit zur Reduzierung des Kraftstoffverbrauchs dar [19]. Die zusätzlichen Vorteile eines Brennstoffzellensystems können in Zukunft die negativen Eigenschaften überwiegen und in der Summe die direkten Betriebskosten positiv beeinflussen. Mit den vorgestellten PDh-Konzepten ist ein Einsatz an Bord eines Flugzeugs allerdings nicht anzuraten.

In der Strategie der Luftfahrtindustrie ist mittelfristig der Einsatz leistungsstärkerer Brennstoffzellensysteme für die Substitution derzeitiger Flugzeugsysteme, wie z. B. der APU, vorgesehen. Im folgenden Abschnitt werden zwei Konzepte mit einer Leistung von 100 kW und einer Stromdichte von 4000 A/m² dargestellt. In beiden Konzepten wird eine HT-PEMFC der Firma PEMEAS eingesetzt, sodass eine Befeuchtung der Membran und eine Feingasreinigung überflüssig sind [128]. Die Konzepte unterscheiden sich lediglich in der Brenngasreformierung. In Konzept 5 wird ein autothermer Reformier eingesetzt, während in Konzept 6 die PDh-Reformierung verwendet wird. Für den Aufbau der Modelle wurde die bereits beschriebene Modellbibliothek mit standardisierten Schnittstellen verwendet. Die Simulationsergebnisse sind in Tabelle 5-6 dargestellt.

	Konzept 5 ATR-Reformierung	Konzept 6 PDh- Reformierung	Einheit
Leistung	100	100	kW
Zellspannung	0,6	0,62	V
Treibstoff	0,044 25	6,75 3800	mol/s kg/h
Wasserstoff	1,03 7,4	1,01 7,2	mol/s kg/h
Wasserproduktion	0,356 23	0,844 55	mol/s kg/h
Brennstoffzellenwirkungsgrad	40	41,1	%
Systemwirkungsgrad (ISA-Bodenbedingungen)	28	18	%

Tabelle 5-6 Simulationsergebnisse der Konzepte 5 und 6

Der wesentliche Unterschied der beiden Konzepte besteht im Treibstoffbedarf. Bei gleicher elektrischer Leistung an der Brennstoffzelle von 100 kW ist für das System mit partieller Dehydrogenierung ein Treibstoffstrom von 6,75 mol/s bzw. 1,055 kg/s nötig, wohingegen der Treibstoffstrom beim System mit autothermer Reformierung lediglich 4,41 e-2 mol/s bzw. 7 g/s beträgt. Der große Unterschied folgt aus der 15-prozentigen Umsetzung des PDh-Systems, d. h., aus ca. jedem sechsten Molekül C₁₁H₂₄ (Undekan als Kohlenstoffmolekül für Kerosin) wird ein Molekül Wasserstoff reformiert. Bei der Reaktion der autothermen Reformierung werden hingegen von 1 mol Undekan 12,99 mol Wasserstoff extrahiert. Zusätzlich wird bei der Kohlenmonoxidumwandlung der Shift-Reaktion Wasserstoff erzeugt, sodass sich für das gesamte Reformierungssystem eine Wasserstoffausbeute von 23,49 mol aus 1 mol Kohlenwasserstoff C₁₁H₂₄ errechnet. Hierbei geht allerdings nicht der gesamte Wasserstoff aus dem Kohlenwasserstoff, sondern ein Teil aus dem an den Reaktionen beteiligten Wasser hervor. In Abbildung 5-16 sind sämtliche Stoffströme des ATR-Konzepts dargestellt.

Weitere Unterschiede bestehen in der Leistung, die zur Erwärmung des Treibstoffs zuzuführen ist, sowie in der nötigen Heizleistung für die endotherme Reaktion der partiellen Dehydrogenierung. Bei dem System mit autothermer Reformierung werden die Erwärmung und Verdampfung des Treibstoffs durch die Beimischung des heißen Wasserdampfs erreicht. Für autotherme Bedingungen erfolgen die Erwärmung und Verdampfung des Wassers allein durch die Prozesswärmerückführung. Bei dem System mit partieller Dehydrogenierung erfolgt ebenfalls eine Prozesswärmerückführung. Es muss dennoch eine Heizleistung aufgebracht werden, um den Treibstoff auf die Reaktionstemperatur für die Wasserstoffabspaltung zu

Gesamtkonzeptevaluierung

erwärmen. Zusätzlich ist für den Ablauf der endothermen Abspaltungsreaktion eine Heizleistung von 131 kW erforderlich.

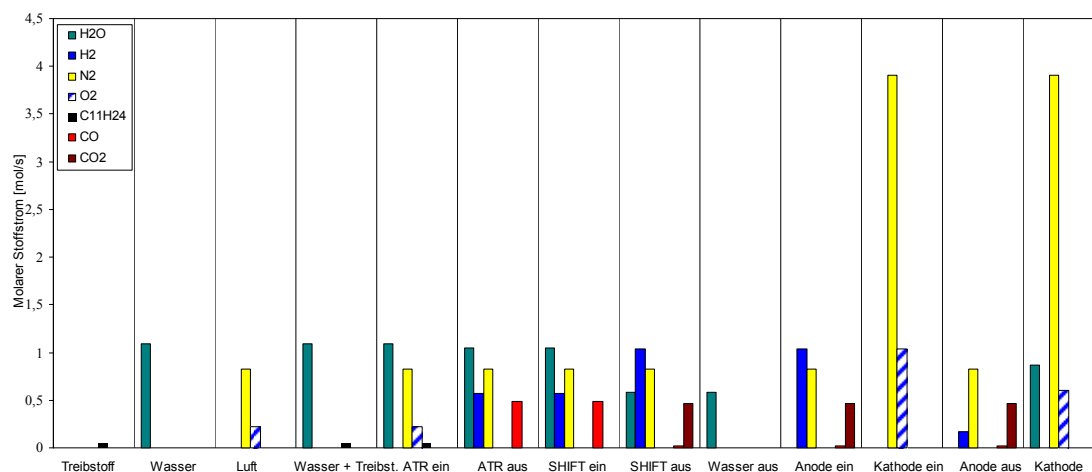


Abbildung 5-16 Stoffströme im ATR-Konzept

Dem gegenüber steht die Tatsache, dass die benötigte Wasserstoffmenge bei dem System mit partieller Dehydrogenierung geringer als beim Konzept mit ATR ist. Das bei der autothermen Reformierung entstehende Reformat beinhaltet neben dem gewünschten Wasserstoff noch weitere unerwünschte Bestandteile, wie Kohlenmonoxid, Kohlendioxid und Stickstoff. Dieses ist bei dem System mit PDh nicht der Fall. Hier wird der Brennstoffzelle reiner Wasserstoff zugeführt. Da der Systemdruck in beiden Systemen gleich groß ist, liegt bei dem System mit partieller Dehydrogenierung ein höherer Wasserstoffpartialdruck vor. Daraus resultiert eine leichte Erhöhung der Zellspannung. Um die gleiche elektrische Leistung der Brennstoffzelle zu erzielen, wird somit beim System mit partieller Dehydrogenierung ein geringerer elektrischer Strom benötigt. Der Strom hängt linear mit dem umgesetzten Wasserstoff zusammen [siehe Gleichung (5-4)]. Daraus folgen ein geringerer Wasserstoffstrom für das System mit partieller Dehydrogenierung sowie ein höherer Wirkungsgrad der Brennstoffzelle. Die benötigten Leistungen zum Vorwärmen des Treibstoffs und zur Aufrechterhaltung der PDh-Reaktion überwiegen bei Weitem den geringeren Wasserstoffbedarf des PDh-Systems. Insgesamt führt dies zu einem Unterschied des Systemwirkungsgrads gegenüber dem ATR-System von 10 % (siehe Tabelle 5-6).

Wie bereits erwähnt, ist die Wasserproduktion des Brennstoffzellensystems für den Einsatz an Bord ebenfalls von Bedeutung. Bei dem System mit partieller Dehydrogenierung wird kein zusätzliches Wasser innerhalb des Systems benötigt. Da die verwendete HT-PEM-Brennstoffzelle keine Befeuchtung benötigt, kann das gesamte Produktwasser entnommen werden. Für das PDh-Konzept 6 ergibt sich eine Wasserproduktion von 0,884 mol/s, das entspricht umgerechnet etwa 55 kg/h. Bei dem System mit autothermer Reformierung wird sowohl für den Anteil der Dampfreformierung innerhalb der autothermen Reformierung als auch zur Kohlenmonoxidumwandlung in der Shift-Reaktion Wasser benötigt und umgesetzt. Die Menge an Wasser, die dem System entnommen werden kann, ist somit bei einem System mit autothermer Reformierung geringer; hier wurde ein Stoffstrom von 0,356 mol/s bzw. etwa 23 kg/h ermittelt. Das System mit partieller Dehydrogenierung stellt somit die größere Wassermenge bereit.

Ein weiteres Kriterium zur Beurteilung der Systeme ist die Komplexität. Diese ist bei dem System mit partieller Dehydrogenierung deutlich geringer. Als Edukt geht, abgesehen von der Umgebungsluft für die kathodenseitige Versorgung der Brennstoffzelle, lediglich der Treibstoff ein. Bei dem System mit autothermer Reformierung ist die Zuführung von Luft und Wasser in einem genau zu regelnden Verhältnis nötig. Darüber hinaus besteht die Brenngaserzeugung bei dem System mit partieller Dehydrogenierung nur aus zwei Prozessschritten, nach der Reaktion der partiellen Dehydrogenierung ist lediglich noch eine Stoffauftrennung vorzunehmen. Das System mit autothermer Reformierung beinhaltet neben dem eigentlichen Reformier zusätzlich den Shift-Reaktor zur Kohlenmonoxidumwandlung sowie einen Wasserabscheider vor der Anode der Brennstoffzelle. Bei dem ermittelten Kohlenmonoxidgehalt des Reformats von weniger als 1 % kann allerdings auf eine Feingasreinigung bei Einsatz einer HT-PEMFC mit einem maximal zulässigen Kohlenmonoxidanteil des Anodengases von 1 %

5.2 Realisierung des Simulationskonzepts

verzichtet werden. Aus der geringen Systemkomplexität des PDh-Systems resultiert eine Erhöhung des Robustheitsgrads. Dies hat wiederum eine Reduzierung der Wartungsaufwendungen zur Folge.

Um diese Beurteilung abzuschließen, soll noch einmal auf die Unterschiede der Systeme in Bezug auf die Schadstoffemission eingegangen werden. Bei dem System mit autothermer Reformierung sind im Reformat und somit auch im Abgas der Brennstoffzelle sowohl Kohlendioxid als auch Kohlenmonoxid enthalten, wohingegen bei dem System mit partieller Dehydrogenierung im Abgas der Brennstoffzelle weder Kohlenmonoxid noch Kohlendioxid enthalten sind. Der Brennstoffzelle wird reiner Wasserstoff zugeführt. Das Abgas besteht hier aus Wasser, Luft sowie nicht umgesetztem Wasserstoff.

Abschließend bleibt festzuhalten, dass ein Brennstoffzellensystem mit partieller Dehydrogenierung ungeachtet der vorhandenen systembedingten Vorteile des reineren Reformats sowie der geringeren Systemkomplexität derzeit aufgrund des deutlich geringeren Systemwirkungsgrads gegenüber einem System mit autothermer Reformierung nicht zu bevorzugen ist.

Die unterschiedlichen Umgebungsanforderungen an eine konventionelle Gasturbinen-APU führen zu Systemwirkungsgraden um 15 %. Zusätzlich verursacht die APU 20 % der Luftschadstoffemissionen und 50 % der wartungsbedingten Verspätungen [102]. Diese Gründe sprechen dafür, die derzeitige APU durch ein Brennstoffzellensystem zu ersetzen. In Tabelle 5-7 sind die Simulationsergebnisse eines PDh-Konzepts mit einer elektrischen Leistung von 83 kW und einer HT-PEMFC den Eigenschaften der APS 3200 APU des Unternehmens Hamilton Sundstrand gegenübergestellt [62]. Wie bereits für die Konzepte 3 und 4, wurde der optimale Betriebspunkt der Brennstoffzelle in Bezug auf die minimale Systemmasse in einer Sensitivitätsanalyse ermittelt. Der höhere Wirkungsgrad des PDh-Systems gegenüber dem sechsten Konzept (vgl. Tabelle 5-6) ist auf den Einsatz eines Nachbrenners zurückzuführen (vgl. Abbildung 5-10).

	APS 3200 APU	PDh-Konzept	Einheit
Leistung	83	83	kW
Gewicht	140	776	kg
Spezifische Leistung	0,6	0,1	kW/kg
Wirkungsgrad	15	24	%
Inertisierung	entfällt	4,5 460 ($\lambda = 2$)	mol/s kg/h
Wasserproduktion	entfällt	0,86 56	mol/s kg/h
Zellenspannung	entfällt	0,5	V
Druckluft	1,05	entfällt	kg/s

Tabelle 5-7 Vergleich einer konventionellen APU mit einem PDh-System

Es wird deutlich, dass eine direkte Substitution der APU nicht möglich ist, da die Brennstoffzelle einigen Anforderungen nicht gerecht wird. Die APU stellt neben der elektrischen Energie auch pneumatische Energie für den Triebwerksstart und die Druckluftversorgung in Notfallsituationen zur Verfügung. Die Realisierung einer Brennstoffzellen-APU an Bord eines Flugzeuges erfordert z. B. einen elektrischen Start der Triebwerke sowie eine elektrische Klimaanlage, wie sie in der B787 des Flugzeugherstellers Boeing eingesetzt wird. Um die Vorteile des Brennstoffzellensystems, wie z. B. die Tankinertisierung und die Wasserproduktion, nutzen zu können, sind weitere Modifikationen bestehender Systeme nötig. Ein Nachteil, der zum gegenwärtigen Stand der Technik gegen den Einsatz eines Brennstoffzellensystems mit integrierter Reformierung spricht, ist die niedrigere spezifische Leistung gegenüber der APU.

Der Einsatz der Simulation in der frühen Entwicklungsphase ermöglicht es, System-schwachstellen sowie Parameter, die das System im besonderen Maße beeinträchtigen, zu identifizieren. Dieses Vorgehen ermöglicht es bereits in der Konzeptphase, eine System-optimierung zu durchlaufen. Im Falle des PDh-Konzepts wurden die Wärmerückgewinnung sowie die geringe Umsetzung im PDh-Reaktor als Schwachstellen ausgemacht. Aus diesem Grund wird im Folgenden ein verbessertes Konzept beschrieben.

Ein Ansatz zur Verbesserung besteht darin, die Brennstoffzellenabwärme nicht an die Umgebungsluft abzugeben, sondern zur Vorwärmung des Kerosins zu nutzen. Hierfür muss das Kerosin durch den bisherigen Kühlkreislauf der Brennstoffzelle gepumpt werden. Zusätzlich zur Nutzung der Brennstoffzellenabwärme entfallen der Brennstoffzellen-

wärmetauscher, die Pumpe für das Kühlmedium sowie der Lüfter. Des Weiteren lässt sich der PDh-Reaktor optimieren. Der Einsatz eines porösen Katalysators, der durchströmt wird, ermöglicht eine Steigerung der Kerosinumsetzung. Auf Basis dieser Annahmen wurde eine Konzeptalternative für ein 20-kW-PEM-Brennstoffzellensystem mit partieller Dehydrogenierung erstellt. Das spezifische Gewicht der Brennstoffzelle wurde mit $9,75 \text{ kg/m}^2$ aktiver Membranfläche gerechnet, das entspricht dem Optimalwert aus einer Marktanalyse von 40 Brennstoffzellenstacks. Die ermittelte Systemmasse von 103 kg stellt nahezu eine Halbierung der Systemgewichte von Konzept 3 und 4 dar. Dieses Konzept übertrifft sogar die Konzepte mit Druckflaschen. Allerdings sei an dieser Stelle darauf hingewiesen, dass für das Umgebungsszenario „Heißer Bodenfall“ nicht die gesamte Abwärme an das Kerosin abgegeben werden kann. Dies führt zu einer Verminderung der Brennstoffzellenleistung, die durch den Einsatz einer HT-PEMFC verhindert werden kann. Die Ergebnisse der Simulation zeigen, dass mit den getroffenen Annahmen ein Brennstoffzellenkonzept für den kontinuierlichen Betrieb an Bord eines Flugzeugs geeignet ist.

In diesem Kapitel wurde gezeigt, dass durch den Einsatz von Simulationstechniken in der frühen Entwicklungsphase die Entscheidungsqualität verbessert wird. Konzeptalternativen können zeitnah und mit geringem Ressourceneinsatz quantitativ verglichen werden, daraus resultiert die Weiterentwicklung der den Anforderungen am besten entsprechenden Konzeptalternative. Der Modellierungsaufwand der in dieser Phase verwendeten Architekturmodelle ist gering. Eine Modularisierung der Modelle, wie sie im Vorangegangenen durch den Einsatz einer Brennstoffzellenbibliothek realisiert wurde, sowie standardisierte Modellschnittstellen unterstützen zusätzlich die Konzeptmodellerstellung und erhöhen deren Wiederverwendbarkeit. Mit voranschreitendem Entwicklungsprozess muss sich der Modelldetaillierungsgrad weiter erhöhen, dies ist nötig, um eine kontinuierliche Simulationsunterstützung der Systementwicklung zu gewährleisten. Aus diesem Grund wird im folgenden Kapitel die Umsetzung eines funktionalen Modells der Detaillierungsphase (M5–M6) beschrieben.

5.2.2 Funktionales Modell der Brennstoffzelle

Analog zum Top-down-Prinzip im linken Strang des V-Modells muss für eine kontinuierliche Simulationsunterstützung der Modelldetaillierungsgrad bis zur Komponentenentwicklung stetig steigen. In der frühen Detaillierungsphase werden aus diesem Grund funktionale Modelle benötigt, diese eignen sich im Gegensatz zu den Architekturmodellen für dynamische Systemanalysen und erlauben Stabilitätsuntersuchungen sowie die Analyse des Systemverhaltens auf äußere Anregungen.

In der vorangegangenen Konzeptphase wurden mithilfe der Simulation verschiedene Konzeptalternativen des Brennstoffzellensystems quantitativ bewertet. Für die Weiterentwicklung in der Detaillierungsphase wurde aufgrund der Simulationsergebnisse ein Brennstoffzellensystem, bestehend aus einem autothermen Reformierungsprozess und einer PEM-Brennstoffzelle, ausgewählt. Aus diesem Grund wird im Folgenden die Umsetzung eines funktionalen Modells anhand einer PEM-Brennstoffzelle demonstriert.

Anders als bei den vorangegangenen Architekturmodellen wird die Brennstoffzelle so modelliert, dass die benötigten Stoffströme als Funktion der angeschlossenen Last berechnet werden. Als Eingangsgröße des Modells dient der Strom, die resultierenden Stoffströme werden als Ausgabegrößen des Modells berechnet. Dieses Vorgehen ermöglicht die dynamische Lastanalyse der Brennstoffzelle. Dynamische Langzeiteffekte, wie die Degradation, werden in dem funktionalen Modell nicht berücksichtigt. In den Verhaltensmodellen der nächsthöheren Detaillierungsebene sind diese Effekte zu berücksichtigen. In Abbildung 5-17 ist der Modellaufbau dargestellt. Das funktionale Brennstoffzellenmodell wurde mit der Software Matlab/Simulink erstellt und setzt sich aus drei Blöcken zusammen. In den ersten beiden Blöcken zur Berechnung der Leerlaufspannung und des Arbeitspunktes aus der Polarisationskennlinie werden analytische Lösungen verwendet; hierbei handelt es sich um rein algebraische Gleichungssysteme. In einem dritten Block wird das dynamische Verhalten der Brennstoffzelle mittels Differenzialgleichungen beschrieben.

5.2 Realisierung des Simulationskonzepts

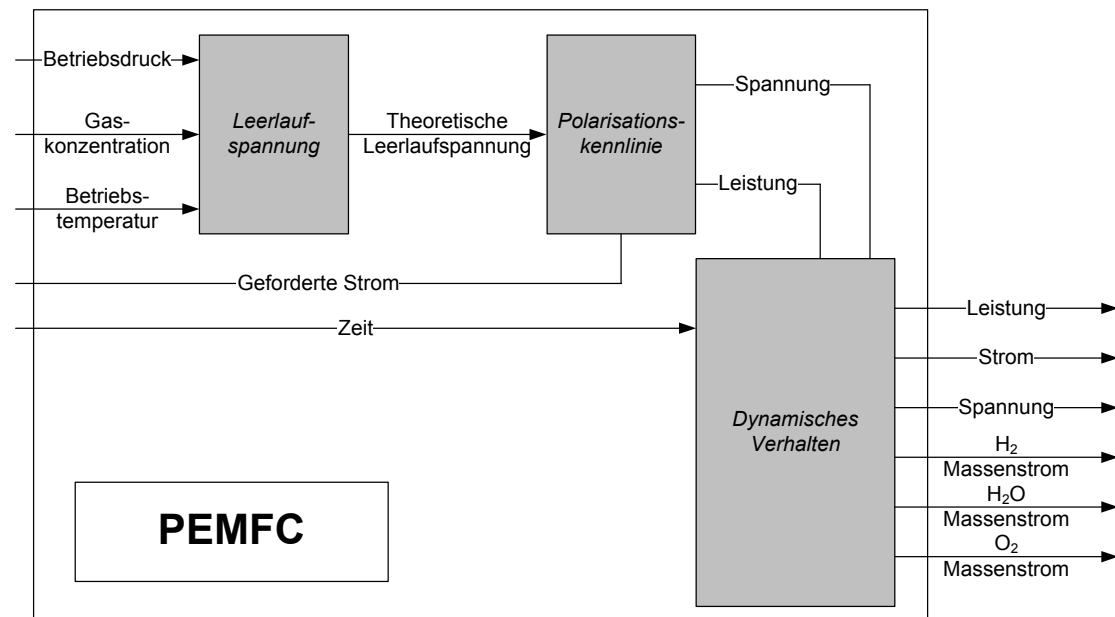


Abbildung 5-17 Dynamisches Brennstoffzellenmodell

Die Berechnung der Leerlaufspannung unter Standardbedingungen liefert in vielen Fällen eine ausreichende Genauigkeit. Für den Betrieb einer Brennstoffzelle an Bord eines Fluggerätes weichen die realen Betriebswerte allerdings signifikant von den Standardbedingungen ab. Aus diesem Grund ist eine Berücksichtigung der Betriebsparameter Druck und Temperatur unumgänglich. In der Gleichung (5-7) wird der Einfluss der Temperatur auf die Leerlaufspannung U deutlich.

$$U_{rev} = \frac{-\Delta \bar{g}}{2 \cdot F} \quad (5-7)$$

Die Gibbssche Energie g , also die tatsächliche nutzbare Energie, setzt sich aus der Reaktionsenthalpie und der Reaktionsentropie zusammen. Diese sind wiederum direkt von der Temperatur abhängig. Ein Anstieg der Temperatur bewirkt damit einen Abfall der Leerlaufspannung. Dabei hat eine Entropieänderung die größte Auswirkung auf den Leerlaufspannungsabfall. In Abbildung 5-18 ist die Leerlaufspannung als Funktion der Temperatur im Bereich von 0 bis 125 °C dargestellt.

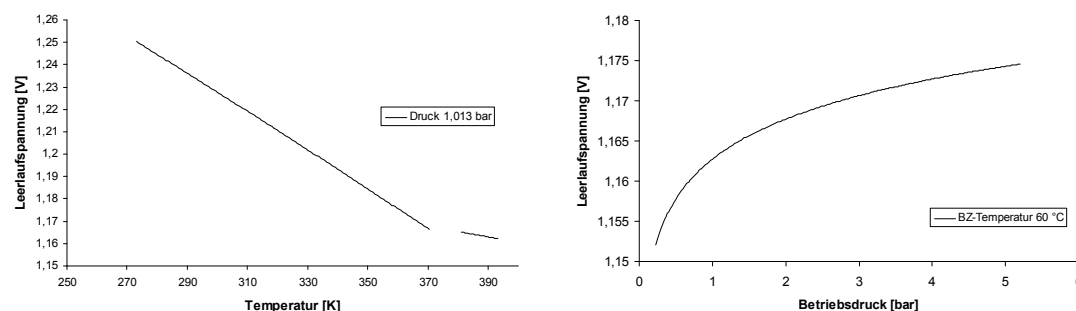


Abbildung 5-18 Temperatur- (links) und Druckabhängigkeit (rechts) der Zellspannung

Um den Einfluss des Betriebsdrucks und der Gaskonzentration zu berücksichtigen, eignet sich die allgemeine Betrachtung des elektrochemischen Elektrodenpotenzials in Abhängigkeit von der Gaskonzentration bei Redox-Reaktionen. Dieser Zusammenhang ist in der nernstschen Gleichung berücksichtigt. Die allgemeine nernstschen Gleichung besitzt zentrale Bedeutung in der Elektrochemie, Galvanik und Elektroanalytik, weil sie die elektrische Größe Spannung (bzw. Elektrodenpotenzial) mit der chemischen Größe Konzentration verbindet [105]. In Abbildung 5-18 ist bei einer konstanten Betriebstemperatur von 60 °C der Druck von 0,2 auf 5 bar erhöht worden. Variiert man den Betriebsdruck der Brennstoffzelle mit positivem

Gradienten, zeigt sich ein Anstieg der Zellspannung. Die Ursache hierfür ist auf den Konzentrationsanstieg mit zunehmendem Druck zurückzuführen. Bei steigender Konzentration stehen mehr Ladungsträger pro Volumen zur Verfügung; dies hat eine Steigerung der Ladungsträgerdichte sowohl auf der Oberfläche der Elektroden als auch durch die Membran zur Folge. Für einen hohen Brennstoffzellenwirkungsgrad ist ein Betrieb bei hohem Systemdruck und niedriger Temperatur anzustreben. Es sei darauf hingewiesen, dass dieser Betriebsbereich nicht für sämtliche Brennstoffzellensysteme einen optimalen Systemwirkungsgrad zur Folge hat. Bei einer Kombination aus Brennstoffzelle und Wärmekraftmaschine ist z. B. eine hohe Brennstoffzellentemperatur empfehlenswert.

Die Berechnung der Brennstoffzellenspannung in Abhängigkeit des geforderten Stroms wird im zweiten Block des Modells vorgenommen (siehe Abbildung 5-17). Mithilfe der Butler-Volmer-Gleichung [vgl. Gleichung (5-2)] wird der Arbeitspunkt auf der Polarisationskennlinie mit der zuvor ermittelten theoretischen Leerlaufspannung berechnet. Zur Ermittlung der fehlenden Koeffizienten der Gleichung wurde die Polarisationskurve einer realen Brennstoffzelle im Laborversuch ermittelt. Für die Aufnahme der Kennlinie wurde eine Luft atmende PEM-Brennstoffzelle mit einer elektronisch steuerbaren Last beschaltet. In einer HIL-Simulation wurden die Steuerungssignale der elektrischen Last mithilfe der Simulationssoftware Matlab/Simulink generiert. Für die Aufnahme der Polarisationskurve muss, ausgehend von offenen Klemmen, der elektrische Lastwiderstand stetig verkleinert werden. Mithilfe der Kennlinie ist es möglich, die flächenspezifische Impedanz der Brennstoffzelle näherungsweise zu ermitteln. Die fehlenden Koeffizienten der Butler-Volmer-Gleichung können mittels Parametervariation iterativ berechnet werden.

Im dritten Block wird das dynamische Verhalten der Brennstoffzelle modelliert. Für die Modellierung des elektrischen Verhaltens existiert ein Ersatzschaltbild der Brennstoffzelle [99]. Diese Ersatzschaltung ist für die Modellierung der Brennstoffzelle allerdings unzureichend genau. Die Schlussfolgerung daraus ist es, die Kennlinien der Laborbrennstoffzelle aufzunehmen und deren Verlauf durch geeignete Funktionen mathematisch zu beschreiben. Hierfür wurde der Spannungsverlauf der Brennstoffzelle in zwei Bereiche eingeteilt: zum einen der Bereich des Spannungsrückgangs bei spontan einsetzender elektrischer Belastung und zum zweiten der des Spannungsanstiegs bei plötzlicher Lastreduktion. Das dynamische Spannungsverhalten in diesen beiden Bereichen wird durch zwei überlagerte Tiefpassfunktionen beschrieben.

$$U(t) = U \cdot \left(\left(K \cdot \left(1 - e^{-\frac{t}{T_1}} \right) \right) + \left((1-K) \cdot \left(1 - e^{-\frac{t}{T_2}} \right) \right) \right) \quad (5-8)$$

Dabei sind T_1 und T_2 die Zeitkonstanten der Tiefpassglieder, und K ist der Gewichtungskoeffizient. Zur Bestimmung der Parameter wurden der Brennstoffzelle mittels HIL-Simulation Lastprofile vorgegeben und die daraus resultierenden Strom- und Spannungsmesswerte aufgezeichnet. Die so gewonnenen Messdaten dienen innerhalb einer in Matlab implementierten Funktion als Zielvorgabe für die Parameterermittlung. Durch Parametervariation werden innerhalb dieser Funktion die drei Parameter iterativ angenähert. In Abbildung 5-19 ist die Sprungantwort des Systems auf einen Lastsprung dargestellt. In diesem Fall wurde der elektrische Widerstand sprunghaft gesenkt. Bedingt durch die kapazitive Impedanz der Brennstoffzelle eilt der Strom der Spannung voraus [93]. Ein plötzlicher Anstieg des Stroms führt damit zu einer verzögerten Spannungsabnahme mit PT1-Charakteristik.

Infolge des verzögerten Spannungsverlaufs und der direkten Stromerhöhung kommt es zu Beginn des Lastwechsels zu einer Leistungsamplitude, gefolgt von einer Absenkung der Leistung, da sich die elektrische Leistung proportional zur Spannung verhält. Die Reaktionszeit der Brennstoffzelle für den verwendeten Typ liegt in Abhängigkeit des Lastsprungs im Bereich von ein bis zwei Sekunden. In einem Brennstoffzellensystem mit Reformierung stellt somit die Brennstoffzelle nicht die zeitkritische Komponente dar. Problematischer ist die Regelung des Reformers. Die Bereitstellung des benötigten Wasserstoffs mittels PDh-Reformer zeichnet sich durch eine höhere Dynamik als die ATR-Reaktion aus, es muss lediglich der Kerosinstrom reguliert werden. In einem autothermen Reformer müssen hingegen der Sauerstoff-, der Kerosin-, der Wasserzufluss sowie die autotherme Reaktion durch die Verhältnisse aus Wasserdampf zu Kohlenstoff und Sauerstoff zu Kohlenstoff geregelt werden.

5.2 Realisierung des Simulationskonzepts

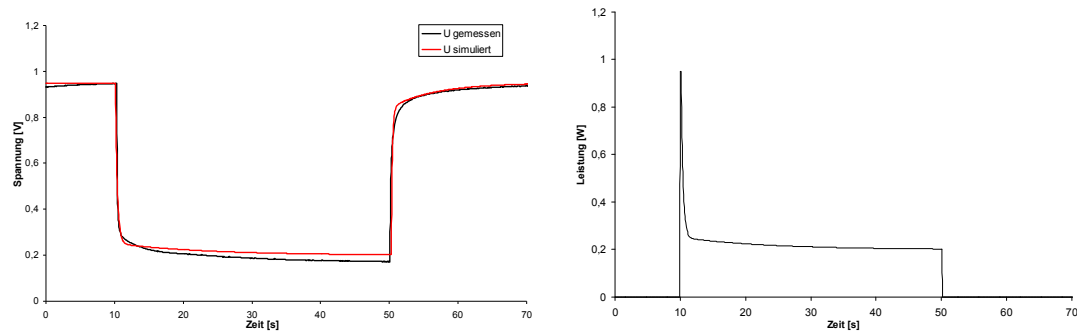


Abbildung 5-19 Spannungs- (links) und Leistungsänderung (rechts) für einen sprunghaftigen Lastwechsel

Dies führt zu einer Verzögerung in der Wasserstoffbereitstellung, die z. B. durch einen Zwischenspeicher kompensiert werden kann. Aus einer Leistungsänderung resultiert zudem ein veränderter Luftmassenstrom, der in einem Kompressor auf Betriebsdruck erhöht werden muss und somit zeitlich verzögert zur Verfügung steht.

Zusätzlich zur dynamischen Brennstoffzellenspannung werden im dritten Block die Wasserstoff-, Sauerstoff- und Wasserströme der Brennstoffzelle in Abhängigkeit von der Leistung sowie der Zellspannung ermittelt [99]. In einer Simulation wurden mit dem Modell Lastprofile analysiert, um so das dynamische Verhalten der Brennstoffzelle bei einem Lastwechsel bewerten zu können. Wie bereits beschrieben, steht bei einer sprunghaftigen Lasterhöhung aufgrund des kapazitiven Verhaltens ein Leistungsüberschuss bereit. Es sei darauf hingewiesen, dass in dem Modell die benötigten Stoffströme zu jedem Zeitpunkt vorausgesetzt werden, d. h., sie stehen ohne Verzögerung zur Verfügung. Dies wird in einem realen System nicht möglich sein (z. B. Leitungslaufzeiten, Flussregelungen, usw.). Für einen Lastsprung mit negativem Stromgradienten wird sich zunächst ein kleinerer Wert als gefordert für die Leistung einstellen. In diesem Fall wird sich die Spannung mit einer Verzögerung von einem niedrigen Potenzial zu einem höheren hin verschieben. In der Simulation wurde zudem deutlich, dass das dynamische Verhalten der Brennstoffzelle vom Arbeitsbereich abhängt. Bei Lastwechseln im höheren Leistungsbereich reagiert sie träger als im Teillastbereich. Je dichter sich ein Lastwechsel an das Leistungsmaximum der Brennstoffzelle annähert, umso langsamer reagiert diese.

Funktionale Modelle eignen sich für die dynamische Systemanalyse. Im vorgestellten Fall konnte mithilfe eines Brennstoffzellenmodells das Lastwechselverhalten analysiert werden. Zudem ist es möglich, das dynamische Verhalten in Bereichen zu untersuchen, die bei der Verwendung der realen Hardware zu kostenintensiven Schädigungen führen könnten. Zusätzlich ergeben sich Zeitersparnisse gegenüber realen Messungen, sodass wesentlich mehr Betriebszustände mithilfe der Simulation untersucht werden können. Für den Einsatz der Modelle in den Testphasen ist auf deren Echtzeitfähigkeit zu achten. Zusätzlich wurde gezeigt, wie Modelle durch den Einsatz von HiL-Simulationen verifiziert werden können. Um den Entwicklungsprozess in allen Phasen durch den Einsatz von Simulationstechniken zu unterstützen, ist es nötig, die funktionalen Modelle weiter zu detaillieren. Aus diesem Grund wird im folgenden Kapitel eine gestalterorientierte Simulation beschrieben.

5.2.3 Gestalterorientierte Simulation eines Brennstoffzellensystems

Im Vergleich zur Systemsimulation werden für die Durchführung einer gestalterorientierten Simulationsstudie (vgl. Kapitel 7.2) erheblich mehr Eingangsparameter und Randbedingungen benötigt. Zudem erfordert sie einen erhöhten Zeit- und Ressourcenaufwand. In der frühen Systementwicklungsphase sind viele Parameter sowie eine dreidimensionale Systemgeometrie noch nicht verfügbar, sodass die Richtigkeit der Ergebnisse, z. B. einer CFD-Simulation, in diesem Stadium fragwürdig ist, da sie auf lückenhaftem Datenmaterial basieren. Der benötigte hohe Modelldetaillierungsgrad und der damit verbundene Ressourceneinsatz einer gestalterorientierten Simulation in der Konzeptfindung (M4–M5) überschreitet bei Weitem ihren Nutzen. Aus diesen Gründen ist die gestalterorientierte Simulation für den Einsatz in frühen Entwicklungsphasen von Flugzeugsystemen ungeeignet. In dieser Phase müssen verschiedene Konzeptalternativen mithilfe der Systemsimulation (vgl. Kapitel 4.1.2) analysiert werden. Hierfür eignen sich Architekturmodelle mit geringer Detaillierung, deren Erstellung einen geringen Ressourceneinsatz erfordert. Die Systemsimulation wird zunächst

zur Festlegung des Konzeptes eingesetzt. In der Detaillierungsphase wird anschließend die Entwicklung dieses Konzeptes vorangetrieben. In dieser Phase ist der Einsatz der gestaltorientierten Simulation zur Analyse verschiedener Systemeigenschaften (z. B. Wärmehaushalt, strukturelle Stabilität, Strömungsverhalten usw.), die nicht von der Systemsimulation berücksichtigt werden, anzuraten. Mit voranschreitender Entwicklung stehen immer mehr Systeminformationen sowie Ergebnisse vorangegangener Systemsimulationen, die als Randbedingungen bzw. als Eingangsparameter für eine gestaltorientierte Simulation dienen, zur Verfügung, sodass die Richtigkeit der Simulationsergebnisse gewährleistet werden kann. Zusätzlich stehen in der Detaillierungsphase (M5–M6) dreidimensionale Geometrien zur Verfügung. Aufgrund des hohen Rechenaufwands ist der Einsatz von gestaltorientierten Simulationen für Echtzeitanwendungen zum gegenwärtigen Zeitpunkt nicht geeignet. Zwar werden z. B. in der Formel 1 bei BMW-Sauber Echtzeitsimulationen der Fahrzeugumströmung durchgeführt, hierfür sind allerdings extrem rechenstarke Computercluster erforderlich [100]. Weiterhin ist die Echtzeit-CFD-Simulation ein Thema in der Medizintechnik. Bei Operationen an der Nasenschleimhaut wird z. B. während des Eingriffs die Auswirkungen der Änderung an der Nasensecheidewand auf die Luftströmung in der Nase mittels Echtzeit-CFD-Simulation analysiert. In der Systementwicklung überschreiten die Anschaffungs- und Betriebskosten der hierfür benötigten Hardware bei Weitem ihren Nutzen. Aus den genannten Gründen ist der Einsatz gestaltorientierter Simulationen von der Detaillierungsphase bis zur Spezifikation, also zwischen M5 und M7, anzuraten.

Die CFD-Analyse eines Brennstoffzellensystems für den Einsatz an Bord eines Flugzeugs dient in diesem Kapitel als Anwendungsbeispiel für den Einsatz von gestaltorientierten Simulationen im Systementwicklungsprozess. Dabei werden Ergebnisse einer Systemsimulation als Eingangsparameter und Randbedingungen für die CFD-Simulation verwendet. In einem Iterationsprozess zwischen CFD- und Systemsimulation werden die jeweiligen Simulationsergebnisse zwischen den Werkzeugen so lange iteriert, bis eine ausreichende Genauigkeit erreicht wird. Für die lose Kopplung zwischen den zwei Simulationsprogrammen wird ein standardisiertes XML-Austauschformat verwendet (siehe Kapitel 5.3.1).

Im Rahmen der Brennstoffzellensystementwicklung wurden die Integrationsaspekte eines Brennstoffzellensystems an Bord eines zukünftigen Flugzeuges mittels CFD-Simulation untersucht. Hierfür wurden die Wechselwirkungen zwischen den Systemkomponenten sowie die Einflüsse der im Integrationsraum vorherrschenden Umgebungsbedingungen berücksichtigt. Basierend auf diesen Untersuchungen wurden Aussagen über eine in Bezug auf den Systemwirkungsgrad geeignete Systemintegration getroffen. Das betrachtete Brennstoffzellensystem ist redundant aufgebaut und stellt eine elektrische Nettoleistung von zweimal 250 kW bereit. Es ersetzt die konventionelle APU und soll neben der Notfallversorgung eine kontinuierliche Energieversorgung gewährleisten. In Abbildung 5-20 ist der Systemaufbau dargestellt. Aus Gründen der eingeschränkten Wasserstoffverfügbarkeit wurde die Reformierung des Wasserstoffs aus dem an Bord mitgeführten Kerosin betrachtet.

Das Brennstoffzellensystem besteht aus einem Reformer, einem Brennstoffzellenstapel, einer Verdichtereinheit, einem Nachbrenner und einer Turbine. Der Flugzeugtreibstoff Jet-A1 wird mittels einer Pumpe in den Reformer geleitet. In einem autothermen Reformierungsprozess (siehe Kapitel 5.1) wird das Kerosin durch die Zugabe von Luft und Wasser in ein wasserstoffreiches Brenngas umgewandelt. Anschließend erfolgt in den Mittel- und Niedertemperatur-Shift-Reaktoren sowie in den selektiven katalytischen Oxidationen (Abk. PROX; engl.: Preferential Oxidation) die Gasreinigung. Für weitere Informationen zur Gasreinigung sei auf folgende Literatur verwiesen [90]. Die Temperatur im Reformer nimmt kontinuierlich vom ATR bis zur PROX-2 ab, sodass die Prozesswärme zum Vorwärmen des Wassers eingesetzt werden kann. Nach Abkühlen des Reformatgases in einem Luftwärmeübertrager wird das darin enthaltene Wasser in einem Entfeuchter abgeschieden und einem Reservoir zugeführt. Der Reformatstrom wird der Brennstoffzelle anodenseitig zugeführt. Überschüssiger Wasserstoff wird in einem katalytischen Nachbrenner mit dem Kathodenabgas der Brennstoffzelle oxidiert. Die hierbei entstehende Wärme wird in einer Turbine in mechanische Energie umgewandelt und treibt damit den ersten Verdichter an.

5.2 Realisierung des Simulationskonzepts

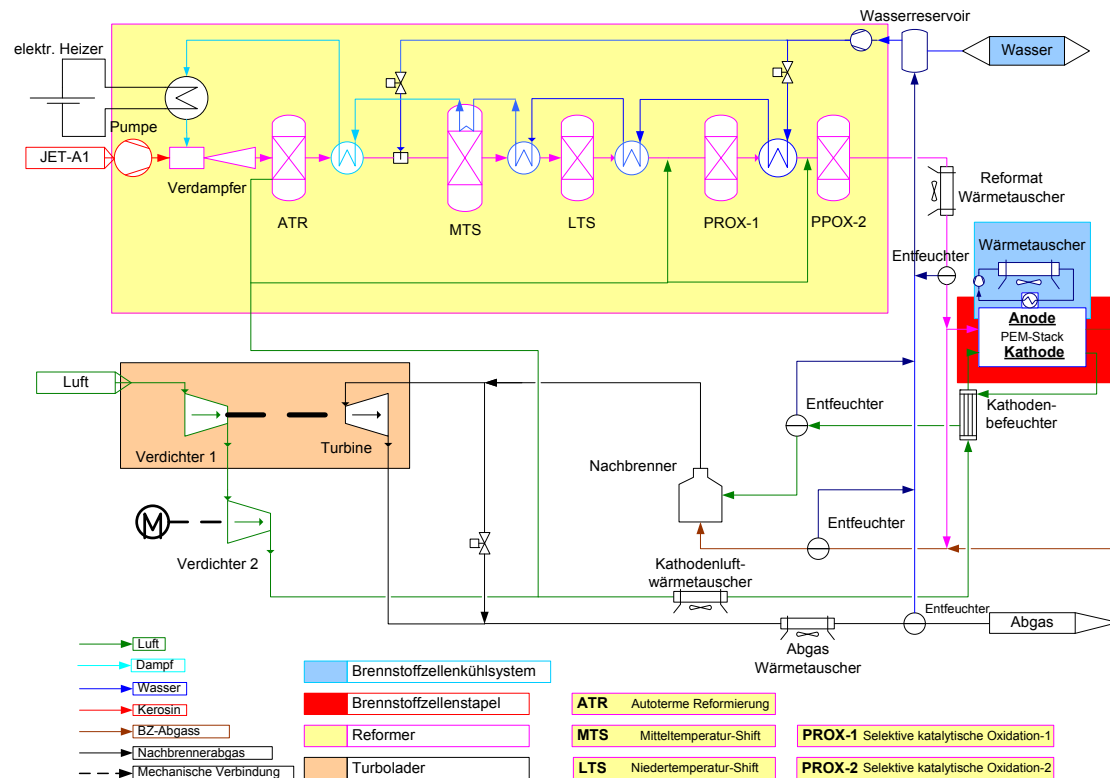


Abbildung 5-20 Aufbau des Brennstoffzellensystems [96]

Ein zweiter Verdichter, der von einem Elektromotor angetrieben wird, bringt den Luftstrom auf den geforderten Systemdruck von 1,4 bar. Die komprimierte Luft wird dem Reformer und der Brennstoffzelle zugeführt. Der Kathodenzuluftstrom muss hierfür in Abhängigkeit von den Umgebungsbedingungen gekühlt werden. Die relative Luftfeuchtigkeit muss zwischen 80 % und 100 % liegen, um einen optimalen Betrieb der Brennstoffzelle zu gewährleisten (vgl. Kapitel 5.1). Aus diesem Grund ist der Brennstoffzelle ein Kathodenluftbefeuchter vorgeschaltet. Der Turbinenabgasstrom wird anschließend in einem Abgaswärmeübertrager so weit gekühlt, bis das kondensierte Wasser mittels Entfeuchter abgeschieden und dem Wasserreservoir zugeführt werden kann. Das im Reservoir gespeicherte Wasser steht für den Reformationsprozess und anderen Flugzeugsystemen als demineralisiertes Grauwasser zur Verfügung.

Um die Interaktionen zwischen dem Integrationsraum und dem Brennstoffzellensystem mittels CFD-Simulation untersuchen zu können, müssen folgende Randbedingungen bekannt sein:

- Geometrie des Integrationsraums
- Szenarien (Betriebs- und Einsatzparameter)
- Systemparameter (zum Teil Ergebnisse der Systemsimulation)

Als Integrationsraum wurde die Sektion 19, das konisch zulaufende Flugzeugende, hinter der Druckkalotte (1) gewählt. In konventionellen Passagiermaschinen ist in diesem Bereich die Gasturbinen-APU installiert. Als Referenzflugzeug dient ein Airbus A330-300. In Abbildung 5-21 ist die dreidimensionale Integrationsgeometrie dargestellt.

Das Brennstoffzellensystem besteht aus zwei redundanten Einzelsystemen und ist symmetrisch zur ZX-Ebene des Flugzeugkoordinatensystems angeordnet. Die vier Brennstoffzellenstapel (2), die Abgaskühler (3), die Abgasentfeuchter (4), die Wechselrichter (5), die Reformatkühler (6) sowie die Reformer (7) befinden sich hinter einem Brandschott (8); während die Systemkomponenten Nachbrenner (9), Verdichter-Turbineneinheit (10), elektrischer Verdichter (11) und Kathodenluftkühler (12) sich vor dem Brandschott (8) über dem Höhenleitwerkskasten (13) angeordnet sind. Die zwei Wärmetauscher (1) der Brennstoffzellenstapel befinden sich in einem mit Außenluft versorgten Kühlkanal (13). Die Wasserreservoirs (14) befinden sich in Flugrichtung hinter dem Kühlkanal (15). Belüftet wird der Integrationsraum durch vier Grenzschichteinlässe.

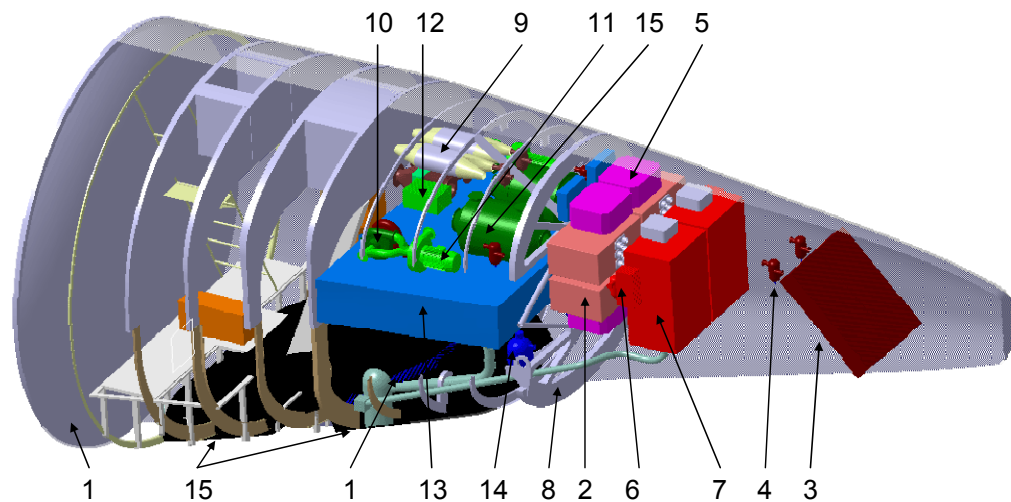


Abbildung 5-21 Integrationsraum des Brennstoffzellensystems [96]

Als weitere Randbedingungen werden die Flugzeugumgebungsbedingungen benötigt. In Tabelle 5-8 sind drei Szenarien enthalten, die in der CFD-Analyse berücksichtigt werden sollen.

Szenario	Flughöhe [ft]	Druck [Pa]	Temperatur [K] Luft/Kerosin/Wasser
1	-2000	108800	329/308/308
2	43000	16230	201/245/308
3	0	101325	288/298/308

Tabelle 5-8 Betriebsparameter des Brennstoffzellensystems

In den ersten zwei Szenarien werden Extremwerte des Flugbereiches berücksichtigt, während das dritte Szenario die ISA-Standardbedingungen enthält und somit als Referenzszenario dient. Die Brennstoffzellen werden in allen drei Szenarien unter Vollast betrieben.

Um das Brennstoffzellensystem in der numerischen Strömungssimulation abbilden zu können, müssen weitere Randbedingungen der Systemgrenzen in die CFD-Simulation eingepflegt werden. Hierzu gehören thermodynamische Transferrandbedingungen, wie z. B. die Wärmeströme über die Komponentenwandung sowie die Betriebstemperaturen der Komponenten. Um die fehlenden Systemparameter, die zu diesem Projektzeitpunkt noch nicht verfügbar waren, zu ermitteln, wurde eine Systemsimulation durchgeführt. Hierbei wurde das Systemmodell, beginnend mit der Leistungsanfrage an die Brennstoffzellen, sequenziell durchgerechnet. Die Leistungsvorgabe dient somit zur Berechnung der Fluidströme in der Brennstoffzelle, diese dient wiederum zur Berechnung der Komponentenparameter des Brennstoffzellenwärmetauschers, des Nachbrenners, des Verdichters und des Reformers (siehe Abbildung 5-22). Für jede Systemkomponente wurden auf diese Weise die jeweiligen Wärmeströme und Betriebstemperaturen ermittelt.

Das allgemeine Vorgehen für die Durchführung der CFD-Simulation ist in Abbildung 5-23 dargestellt. Die Basis der gestaltorientierten Simulation bildet eine dreidimensionale CAD-Geometrie. In diesem Fall wurde die Geometrie mithilfe des CAD-Programms CATIA-V5 erstellt. Die Geometriedaten wurden im Standard for the Exchange of Product Model Data-Format (STEP), einem CAD-Datenformat nach ISO 10303, abgelegt und können direkt in GAMBIT, ein Programm zur Vernetzung zwei- und dreidimensionaler Geometrien, importiert werden. Anschließend wird die Geometrie mit einem Rechenetz diskretisiert. Das diskretisierte Strömungsgebiet wird anschließend als Volumennetz in CFD-Programm Fluent eingelesen und unter Zuhilfenahme der Randbedingungen aus der Matlab/Simulink-Systemsimulation numerisch gelöst. Die Ergebnisse der CFD-Analyse dienen wiederum als Eingangsparameter für die Systemsimulation. Ausgehend von gewählten Anfangsparametern

5.2 Realisierung des Simulationskonzepts

wird so lange zwischen den beiden Simulationsprogrammen iteriert, bis ein vorher festgelegtes Konvergenzkriterium erreicht wird.

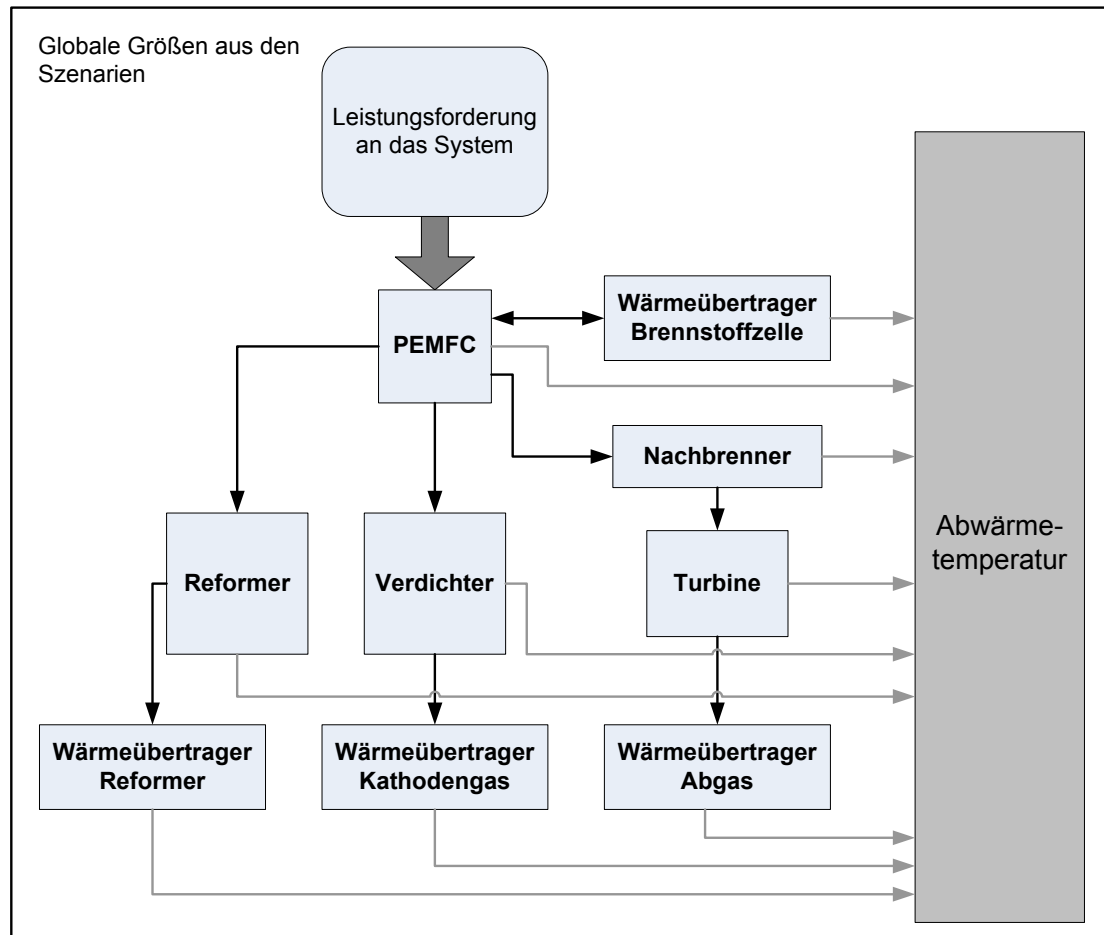


Abbildung 5-22 Schematischer Aufbau der Systemsimulation

Die lose Kopplung der Programme wurde durch den Einsatz eines standardisierten XML-Formats realisiert. Hierfür wurden die proprietären Ergebnisdaten automatisiert in das XML-Standardformat konvertiert. Zur Weiterverwendung in den jeweiligen Simulationsprogrammen wird dieses Format anschließend automatisch in ein proprietäres Format umgewandelt. So kann zum einen die fehlerfreie Weiterverwendung von Simulationsdaten gewährleistet werden, zum anderen ermöglicht die Verwendung eines Standards die problemlose Kopplung mit weiteren Simulationsprogrammen. Die lose Simulationskopplung mittels eines XML-Standardformats wird in Kapitel 5.3.1 detailliert beschrieben. Die Auswertung der Ergebnisse kann je nach Bedarf in Matlab oder Fluent erfolgen.

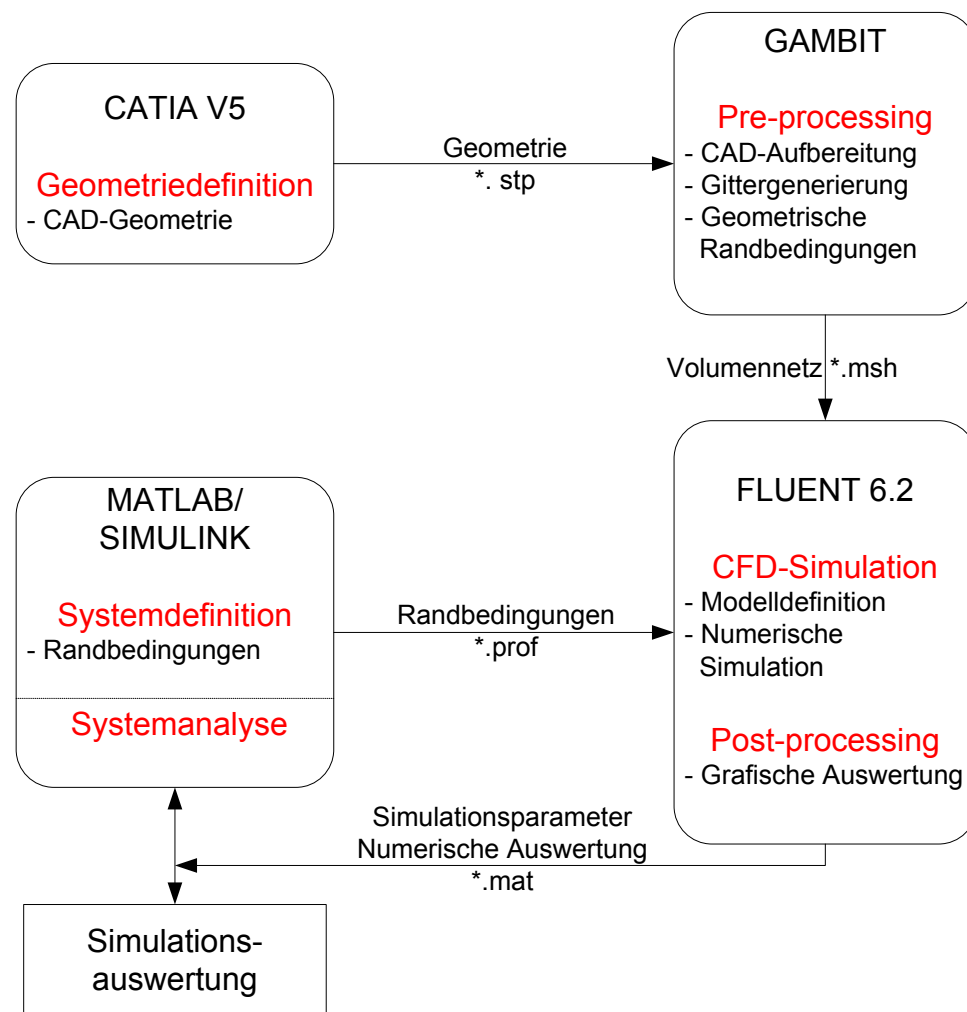


Abbildung 5-23 Lose Kopplung von CFD- und Systemsimulation

5.2.3.1 Ergebnisse der CFD-Simulation

Die in diesem Kapitel beschriebenen Ergebnisse beziehen sich auf den Volllastbetrieb des Systems. In der CFD-Simulation wurde sehr schnell deutlich, dass ohne eine aktive Ventilation in den Integrationssektionen 19.1 vor und 19.2 hinter dem Brandschott der erforderliche Wärmetransport durch freie Konvektionsströmung für die Systemkühlung nicht ausreicht. In der Simulation machte sich dieses durch ein fehlendes Konvergenzverhalten der Residuen sowie einer permanenten Temperaturzunahme im Innenraum bemerkbar. Aus diesem Grund wurden Ventilatoren hinter den Wärmetauschern eingeführt, die den Druckverlust durch den Wärmetauscher kompensieren und eine aktive Luftströmung erzwingen. Die Ventilationsleistung wurde ebenfalls in der Systemsimulation berücksichtigt, in der sie eine Reduzierung des Gesamtsystemwirkungsgrads zur Folge hat. Trotz dieser Maßnahme wurde in der CFD-Simulation deutlich, dass die Wärmetauscher die aus der Systemsimulation hervorgegangenen Wärmeströme nur unzureichend transportierten. Der Grund für dieses Verhalten liegt zum einen in der Unterdimensionierung der Wärmetauscher sowie in der Positionierung der Systemkomponenten im Integrationsraum. In Abbildung 5-24 ist als Beispiel hierfür ein Ergebnis der CFD-Simulation für Brennstoffzellenwärmetauscher im Kühlkanal dargestellt.

Aus den Ergebnissen der Systemsimulation geht hervor, dass der Brennstoffzellenwärmetauscher eine Wärmeleistung von 390 kW pro Brennstoffzellensystem an die Umgebungsluft übertragen muss. Die CFD-Simulation lieferte eine vom Wärmetauscher übertragene Leistung von 301 kW pro Brennstoffzellensystem für das Flugszenario bei einer Umgebungstemperatur von 201 K. Für den Bodenfall wird gerade einmal eine Wärmeleistung von 86 kW angeführt. Dieses ist auf die Positionierung des Wärmetauschers im Kühlkanal zurückzuführen.

5.2 Realisierung des Simulationskonzepts

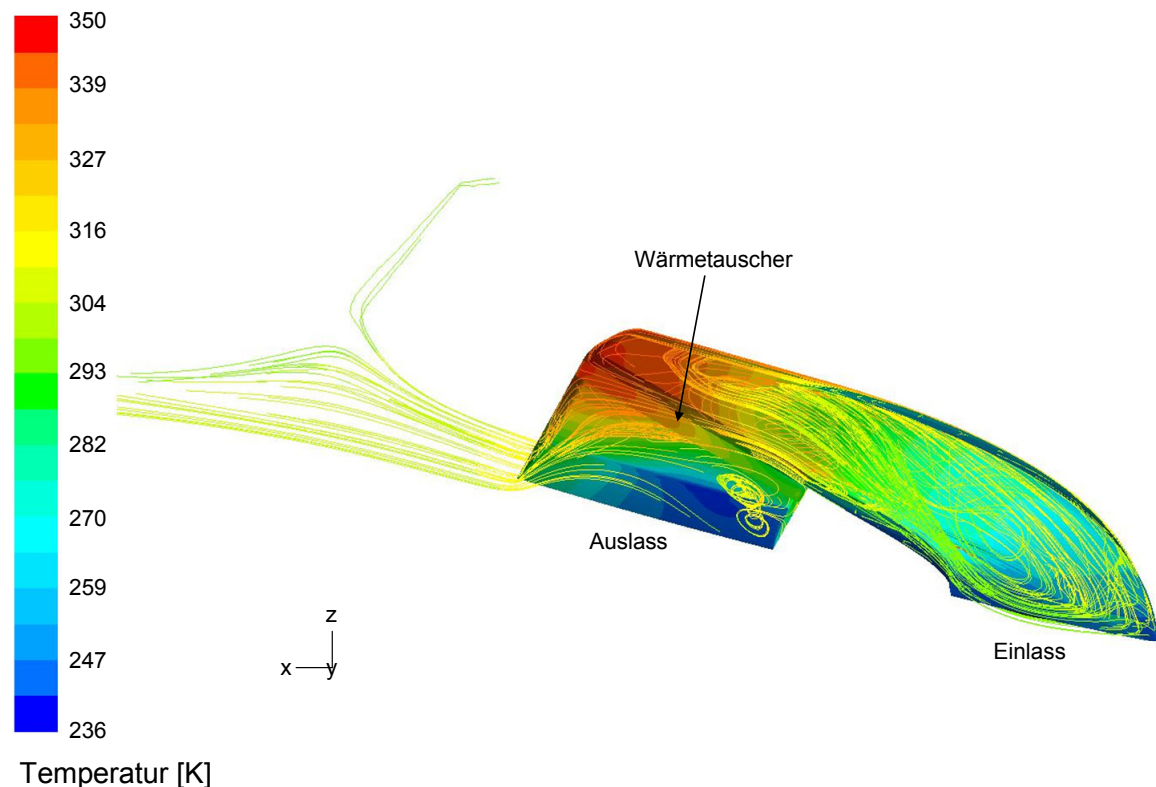


Abbildung 5-24 Kühlkanal mit Brennstoffzellenwärmetauscher (ISA-Szenario)

Zum einen ist der Einbauwinkel des Wärmetauschers zu flach, sodass die Kühlluft nahezu tangential einströmt, sich erwärmt, und durch Konvektionskräfte im Wärmetauscher aufsteigt und eine Zirkulation hervorruft, die bereits vorgewärmte Luft in den Wärmetauscher leitet. Zum anderen ist der Wärmetauscher am oberen Ende des Kühlkanals installiert. Dies hat zur Folge, dass sich warme Luft in diesem Bereich sammelt und den Wirkungsgrad des Wärmetauschers weiter senkt. Ein weiteres Problem stellt die Außengeometrie des Heckkonus dar. Die am Flugzeugrumpf vorbeiströmende Umgebungsluft wird, bedingt durch die konisch zulaufende Form des Rumpfs, verzögert. Dies hat zur Folge, dass der statische Druck auf der Rumpfunterseite nach hinten hin zunimmt. Da der Auslass des Kühlkanals in Strömungsrichtung weiter stromabwärts liegt, wirkt ein relativ erhöhter statischer Druck auf den Auslass. Es kommt zur Rückströmung in den Luftauslass des Kühlkanals.

Ein weiterer Grund dafür, dass die Wärmetauscher nicht die geforderten Wärmeströme an die Luft abgeben, besteht darin, dass sie zum Teil im geringen Abstand zu Wärmequellen wie z. B. dem Nachbrenner oder dem Reformer, installiert sind. In Abbildung 5-25 ist die Wärmeverteilung in Sektion 19.1 für den ISA-Bodenfall dargestellt.

Um die Temperaturverteilung in der Sektion 19.1 darstellen zu können, wurde ein Temperaturbereich von 288 K bis 535 K gewählt. Die Außentemperatur der Nachbrenner liegt deutlich über diesem Bereich und wird in der Abbildung 5-25 daher schwarz dargestellt. Es ist deutlich zu erkennen, dass die Kathodenwärmetauscher aufgrund des geringen Abstands zu den Nachbrennern eine hohe Außentemperatur aufweisen, wodurch ihre Funktionsweise deutlich eingeschränkt wird. Die hohe Nachbrennertemperatur führt zusätzlich dazu, dass die Flugzeughaut, bedingt durch die freie Konvektion, über den Nachbrennern punktuell auf ca. 200 °C erhitzt wird. Dies führt aufgrund der Wärmedehnung zu mechanischen Verspannungen sowie zu einer Abnahme der Materialfestigkeit und ist aus sicherheitstechnischer Sicht inakzeptabel.

Die Temperatur des elektrischen Verdichters liegt über der Temperatur des turbinengetriebenen Verdichters. Dies ist darauf zurückzuführen, dass für die CFD-Simulation die mittlere Temperatur aus Eintritts- und Austrittstemperatur bestimmt wurde. Für den Bodenfall wird der elektrische Verdichter nicht angetrieben, sodass die Eintritts- und Austrittstemperatur gleich der Austrittstemperatur des stromaufwärts liegenden Verdichters sind und somit die mittlere Temperatur über der des turbinengetriebenen Verdichters liegt.

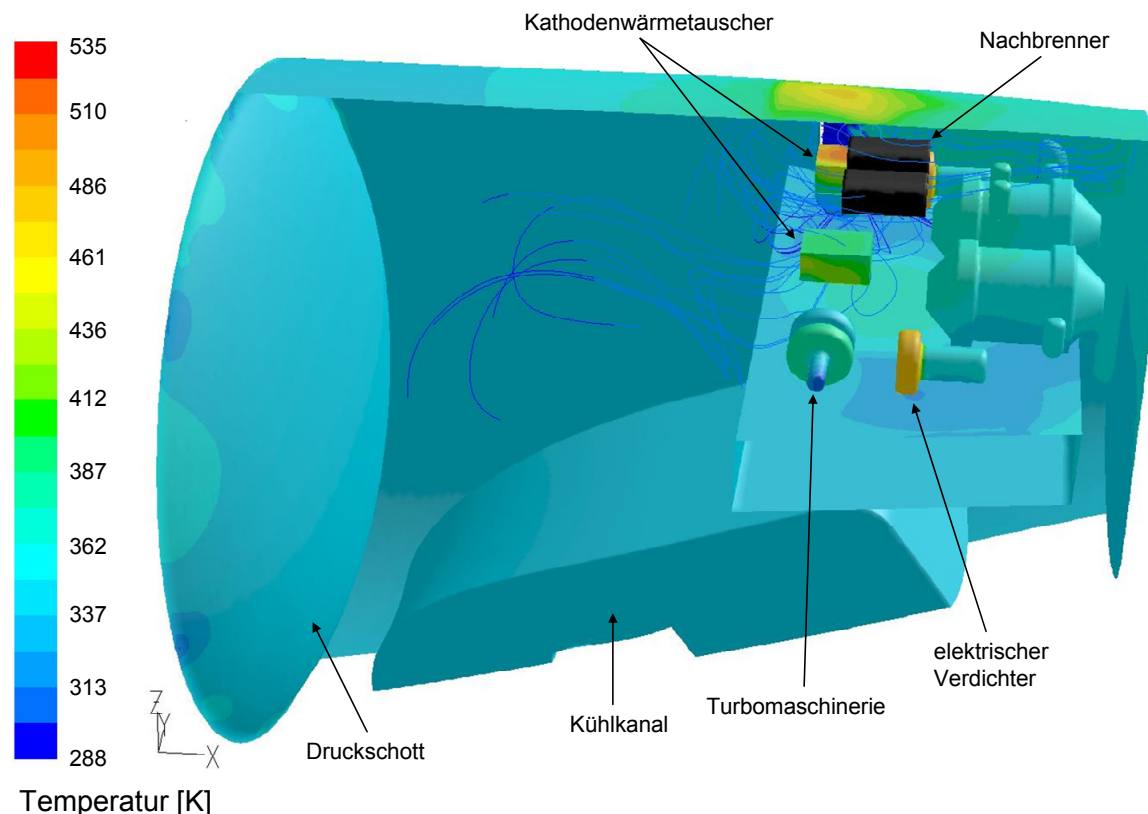


Abbildung 5-25 Wärmeverteilung in der Sektion 19.1 (ISA Szenario)

Die Kombination aus Systemsimulation und CFD-Simulation ermöglichte es, neben den Strömungsverhältnissen und Temperaturverteilungen im Integrationsraum das Systemverhalten zu analysieren. Ausgehend von der Systemarchitektur in Abbildung 5-20 ist zu erkennen, dass durch einen Bypasskanal (engl.: wastegate) Energie aus dem Abgas an der Turbine vorbeigeleitet wird. Dieses Verfahren ist bei Fahrzeugturboladern durchaus üblich; so wird die Verdichterleistung und damit der vom Verdichter erzielte Ausgangsdruck auf einen Maximalwert begrenzt und die Betriebsgrenze des Systems (engl.: surge line) nicht überschritten. In der Systemsimulation wurde deutlich, dass für den Vollastfall in allen drei Szenarien der Bypasskanal geöffnet werden muss, um den Verdichterdruck konstant auf 1,4 bar zu regeln. Unter Vernachlässigung der Strömungsverluste ergeben sich die in Abbildung 5-26 dargestellten Verhältnisse von maximal erzielbarer Turbinenleistung zu genutzter Turbinenleistung.

Zur Bereitstellung des Betriebsdrucks unter Flugzeugumgebungsbedingungen kommt eine zweistufige Radialverdichtereinheit zum Einsatz. Die erste Stufe des Verdichters wird direkt von der Radialturbine, die zweite Stufe von einem Elektromotor angetrieben. Die erste Stufe ermöglicht ein maximales Stufendruckverhältnis von 7. In der Regel werden von den derzeit verwendeten Technologien Stufendruckverhältnisse von ca. 3,5 realisiert; um den Bypassstrom zu minimieren, wurde mit dem theoretisch idealisierten Wert von 7 gerechnet [19]. Bedingt durch den Bypasskanal werden im Bodenstandfall (Szenario 1) 21,7 kW, im Reiseflug (Szenario 2) 113,69 W und unter Bedingungen der ISA-Standardatmosphäre (Szenario 3; vgl. Tabelle 5-8) 25 kW ungenutzt an der Turbine vorbeigeleitet. Zudem benötigt der elektrische Verdichter im Reiseflug eine Leistung von 16,3 kW, da aufgrund des Stufendruckverhältnisses der Druck in der ersten Stufe nur auf 1,1 bar erhöht werden kann.

Durch den Einsatz der Systemsimulation war es möglich, den Wasserhaushalt des Gesamtsystems zu untersuchen. Es wurde deutlich, dass bei einer Leistungsabgabe von 100 % der Abgaswärmetauscher sowie der nachgeschaltete Wasserabscheider für den Wasserhaushalt des Systems nicht benötigt werden (vgl. Tabelle 5-9).

5.2 Realisierung des Simulationskonzepts

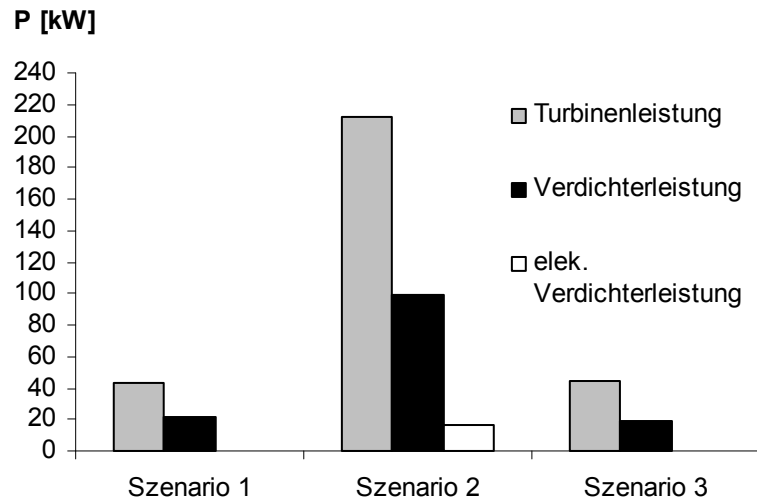


Abbildung 5-26 Leistungsvergleich der Verdichter-Turbineneinheit

Durch den Wegfall dieser Komponenten verringert sich das Systemgewicht und damit indirekt die direkten Betriebskosten des Flugzeugs.

Reformer in [g/s]	Reformer out [g/s]	PEMFC [g/s]	Abgas [g/s]	Summe [g/s]
-67,75	37,04	44	17,87	31,16
-67,75	37,04	44	0	13,29

Tabelle 5-9 Wasserhaushalt des Brennstoffzellensystems

Für die Shift-Reaktion und die Dampfreformierung werden 67,75 Gramm Wasser pro Sekunde benötigt. Die Wasserabscheider nach dem Reformer und der Brennstoffzelle führen einen Wasserstrom von 81 g/s zurück in das Reservoir, während der Abscheider nach dem Abgaswärmetauscher 17,87 g/s produziert. Auch ohne den zusätzlichen Abgaswärmetauscher ergibt sich eine positive Wasserbilanz.

Auf Basis der durch die Simulation gewonnenen Erkenntnisse erfolgte sowohl eine konstruktive als auch eine systemtechnische Optimierung des Brennstoffzellensystems. Als Konsequenz der ineffizienten Arbeitsweise der Verdichter-Turbineneinheit wurde ein alternatives Konzept der Turbomaschinerie konzipiert (siehe Abbildung 5-27). Im Gegensatz zur bisherigen Verdichter-Turbineneinheit wird das gesamte im Abgas enthaltene Energiepotenzial in der Turbine umgesetzt. Die nicht für die Druckerhöhung benötigte Energie wird mittels Generator in elektrische Energie gewandelt und nicht wie bisher über einen Bypass abgeleitet. Der Generator ist als Generator-Motoreinheit ausgeführt, sodass das System ebenfalls über diese Einheit gestartet werden kann.

Bedingt durch die Optimierung der Turbomaschinerie und den zusätzlichen Wegfall des Abgaswärmetauschers und der dazugehörigen elektrischen Ventilationsleistung ergeben sich für das optimierte System folgende Systemeigenschaften. Bedingt durch die zusätzlich erzeugte elektrische Leistung P_{Gen} des Generators erhöht sich die vom System zur Verfügung gestellte Leistung P_{elek} und damit der Systemwirkungsgrad η . Den höchsten Wirkungsgrad von 36,7 % weist das optimierte Konzept im Flugfall auf. Aufgrund des geringen Gegendrucks hinter der Turbine liefert diese eine Leistung von 212 kW. Für die Verdichtereinheit werden 115 kW benötigt, sodass mit einem Wirkungsgradverlust bei der Umwandlung von mechanischer in elektrische Energie 87 kW elektrische Leistung vom Generator zur Verfügung gestellt werden.

Basierend auf den Ergebnissen der CFD-Analyse konnten weitere konstruktive Systemoptimierungen durchgeführt werden. Es wurde deutlich, dass die derzeitigen Lufteinlässe den erforderlichen Luftstrom, der für die Systemkühlung nötig ist, nicht in allen Fällen bereitstellen können. Durch den Einsatz eines NACA-Einlasses kann der benötigte Luftstrom bei kleinstmöglicher Luftwiderstandserhöhung realisiert werden. Bei der Positionierung der Wärmetauscher ist darauf zu achten, dass sie senkrecht zur Strömungsrichtung stehen, dies gilt insbesondere für den Brennstoffzellenwärmetauscher (vgl. Abbildung 5-24). Zudem müssen die Wärmetauscher im unteren Bereich der Sektion 19.1 installiert werden, da sich

warme Luft, hervorgerufen durch die freie Konvektion, im oberen Bereich der Sektion sammelt. Ein Lufteinlass muss in Bereichen hoher statischer Drücke (z. B. unterer Heckkonus) relativ zum Auslass angeordnet sein, sodass eine Rückströmung in den Auslass verhindert wird.

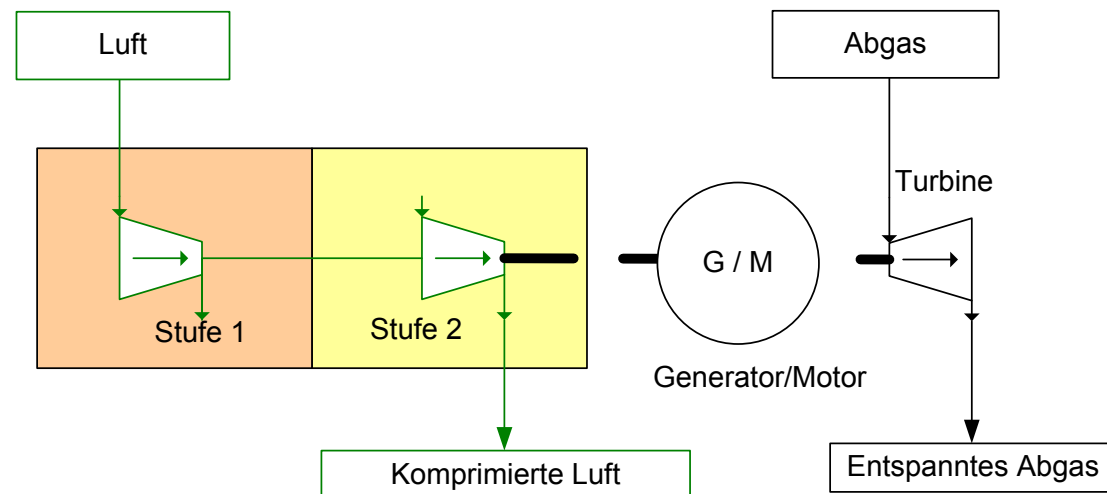


Abbildung 5-27 Alternative Verdichter-Turbineneinheit

Die Systemkomponenten müssen ihrem Temperaturpotenzial entsprechend angeordnet werden (Nachbrenner, Reformer, Turbomaschinerie, Brennstoffzellenstapel, Befeuchter), sodass z. B. der Reformer die Turbomaschinerie und der Brennstoffzellenstapel den Befeuchter vor der einfallenden Wärmestrahlung abschirmen.

Szenario	Altes System		Optimiertes System		
	P_{elek} [kW]	η [%]	P_{elek} [%]	η [%]	P_{Gen} [kW]
1	251	24,7	271	27,3	20
2	261	25,7	365	36,7	87
3	281	27,8	304	30,5	22

Tabelle 5-10 Vergleich der Systemwirkungsgrade und Leistungen

Alle Systemkomponenten, mit Ausnahme der Wärmetauscher, werden hinter dem Brandschott installiert; so kann gewährleistet werden, dass die Funktionsweise der Wärmetauscher nicht durch einfallende Wärmestrahlung beeinflusst wird. Die Simulation hat ergeben, dass im Bereich des Nachbrenners die Flugzeugaußenhaut zusätzlich isoliert werden muss.

Zusammenfassend lässt sich sagen, dass mithilfe der CFD-Simulation konstruktive Systemeigenschaften, wie die Luftströmungen, Temperaturverteilungen im Integrationsraum usw., untersucht werden können. Hierfür müssen allerdings eine Vielzahl von Randbedingungen, Simulationsparametern, Eingangswerten und zusätzlich eine zwei- oder dreidimensionale Geometrie des zu simulierenden Systems bekannt sein. In der frühen Entwicklungsphase ist dies nicht der Fall. Aufgrund der zeitintensiven Modellierung und Simulation eignet sich die gestaltorientierte Simulation ebenfalls nicht für den Vergleich verschiedener Systemkonzepte. Des Weiteren wurde gezeigt, dass sich durch die Kopplung von System- und CFD-Simulation Randbedingungen und Simulationsparameter des jeweils anderen Simulationsprogramms ermitteln lassen und sich damit die Aussagegenauigkeit der Einzelsimulationen deutlich verbessern lassen. Aus diesem Grund wird im folgenden Kapitel die feste Simulationskopplung beschrieben.

5.2.4 Verteilte Simulation des Brennstoffzellensystems

Die verteilte Simulation ermöglicht es, komplexe Systeme, bestehend aus einzelnen Teilsystemen, in einer verteilten Simulationsarchitektur zu simulieren. Aus diesem Grund eignet sie sich für die systemübergreifenden Simulationen im Entwicklungsprozess von der Konzeptphase bis zur Komponentenentwicklung (siehe Abbildung 4-7) sowie für die HIL-

5.2 Realisierung des Simulationskonzepts

Simulation in den Testphasen. Ein Vorteil der verteilten Simulation besteht darin, dass sich, durch den Zusammenschluss lokaler Simulationsrechner zu einem globalen Rechnernetz die Simulationszeit verkürzen lässt.

Der Entwicklungsprozess in der Flugzeugindustrie ist durch eine standortübergreifende Systementwicklung gekennzeichnet. In diesem Zusammenhang bietet die verteilte Simulation die Möglichkeit, Teilsysteme via Internet oder Intranet zu einem komplexen Simulationsnetzwerk zusammenzuschließen und Systeminteraktionen zu analysieren. In diesem Kapitel wird anhand der verteilten Simulation eine Möglichkeit zur Umsetzung der Simulationskopplung für die systemübergreifende Simulation beschrieben (vgl. Abbildung 2-8 Variante 3). Wie schon in den vorangegangenen Kapiteln beschrieben, wird hierfür ein Brennstoffzellensystem modelliert und simuliert. Anschließend erfolgt eine Zusammenfassung über den Einsatz verteilter Simulationen im Systementwicklungsprozess.

Verteilte Simulationen kommen schon seit Jahrzehnten zum Einsatz, um komplexe oder räumlich getrennte Systeme miteinander als Einheit zu verbinden [160]. Es existieren verschiedene Simulationsarchitekturen, wie z. B. die High Level Architecture (Abk. HLA) und die Common Object Request Broker Architecture (Abk. CORBA). Im Gegensatz zu CORBA verfügt die HLA über ein umfangreiches Zeitmanagement. Zusätzlich wurde die vom amerikanischen Verteidigungsministerium entwickelte HLA im Jahre 2000 als IEEE-1516-Standard verabschiedet [39], [40]. Der Standard beinhaltet die Spezifikation für die Erstellung einer verteilten Simulation mithilfe der HLA Run-Time Infrastructure (Abk. RTI) 1516. Die RTI-1.3-NG-Software vom amerikanischen Ministerium für Modellierung und Simulation (Abk. DMSO) steht im Internet für die Weiterentwicklung zur freien Verfügung [38], [155]. Auf Basis dieser RTI entwickelten Unternehmen wie Pitch Technologies AB und MÄAK Technologies kommerzielle Softwareprodukte zur Durchführung von verteilten Simulationen [103], [130]. Die Möglichkeit der einfachen Erweiterbarkeit bestehender Simulationen, der Wiederverwendung von Teilsystemen, des Zeitmanagements, einer freien kostenlosen Verfügbarkeit der RTI Software sowie des IEEE1516 Standards waren ausschlaggebend für die Wahl der HLA. Zudem sieht die HLA Java sowie C++ als Programmierschnittstellen vor und erlaubt damit den plattformunabhängigen Einsatz.

Der prinzipielle Aufbau einer verteilten Simulation mit der HLA ist in Abbildung 5-28 dargestellt. In einer verteilten Simulation kommt es zu Wechselwirkungen zwischen mehreren gleichen oder unterschiedlichen Teilsystemen. Basierend auf dieser Annahme, werden föderalistische Bezeichnungstermini verwendet. Die Simulationsteilnehmer werden im Folgenden als Federates bezeichnet, die sich in einer verteilten Simulation der sogenannten Federation zusammenschließen [38]. Als Simulationsteilnehmer kommen z. B. direkter Programmcode, eingebundene Simulationswerkzeuge oder Hardwarekomponenten infrage. Um eine Simulation zu starten, muss zunächst auf einem Rechner im LAN (Abk. Local Area Network) die RTI Execution (Abk. RTIExec) ausgeführt werden. Sie stellt den zentralen Knotenpunkt dar, über den die gesamte Simulationskommunikation abläuft. Bevor die einzelnen Federates der Federation beitreten können, muss diese zunächst gegründet werden [38]. Diese Aufgabe kann jedem beliebigen Federate implantiert werden. Das Beenden der Federation erfolgt nach einem ähnlichen Schema. Allerdings besitzt nur der letzte noch in der Federation verbliebene Simulationsteilnehmer das Recht, diese zu beenden. Nach der Gründung der Federation können die Federates beitreten. In welcher Reihenfolge dies geschehen soll, ist nicht festgelegt und dem Programmierer überlassen. Bei der Planung der Federation sollte daher ein beliebiges Beitreten der Teilnehmer in Betracht gezogen werden [89].

Nach der Gründung der Federation wird auf dem Hostrechner der RTI Execution die Federation Execution (Abk. FedEx) gestartet und bis zur Beendigung der Federation ausgeführt. Durch den Aufruf der FedEx werden der Federation alle vorher festgelegten Simulationsparameter bekannt gemacht. Diese Parameter werden von dem gründenden Federate übermittelt und liegen in einer vor Beginn der Simulation erstellten Textdatei in HLA-spezifischer Syntax vor. Zu diesen Parametern gehören z. B. die Namen der Federation, der Federates, der Objektklassen, der Attribute usw. Die Attribute dienen zum Datenaustausch zwischen Simulationsteilnehmern, sie können als eine Art Container verwendet werden, in dem beliebige Variablentypen oder Variablenvektoren enthalten sein können. Mit den Interaktionsklassen existiert noch eine weitere Kommunikationsform. Im Gegensatz zu den Attributen müssen die Federates den Interaktionsklassen bei der Gründung der Federation nicht zugeordnet sein. Wenn ein Federate während der Simulation Interaktionen empfängt

oder sendet, kann die dazu notwendige Initiierungsroutine zu jedem beliebigen Zeitpunkt der Simulation ausgeführt werden.

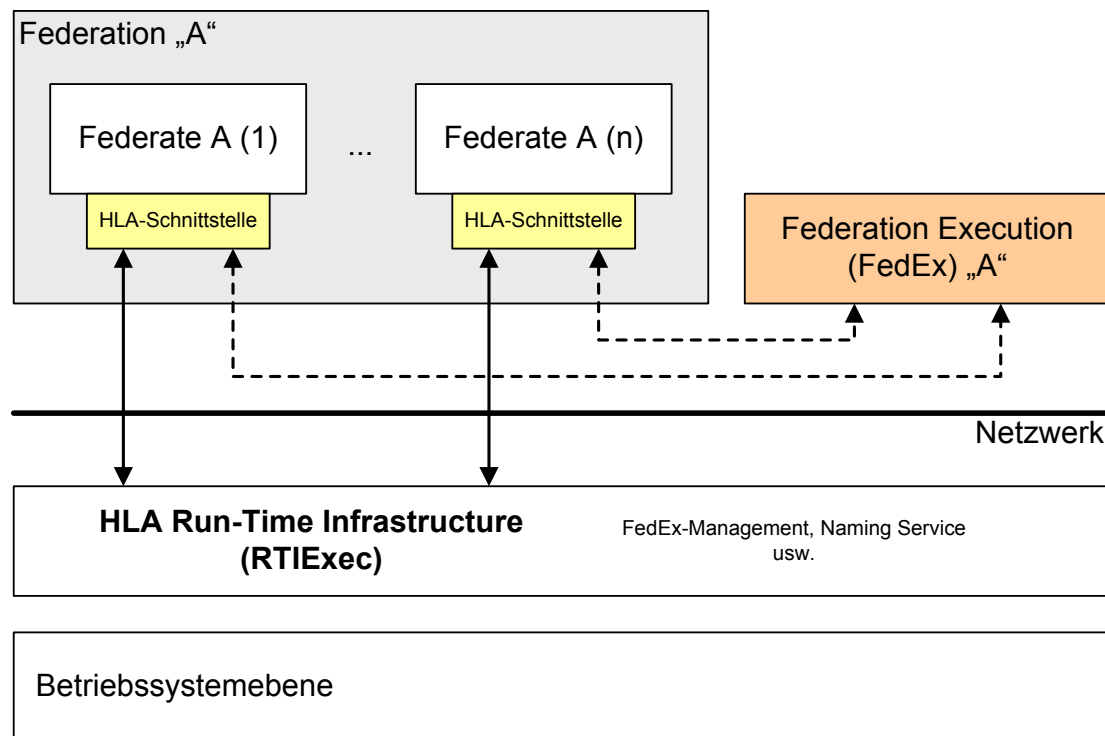


Abbildung 5-28 Aufbau einer verteilten Simulation mit der HLA

Der Nachteil besteht darin, dass bei Interaktionen immer die komplette Klasse übermittelt wird, während bei Objektklassen nur das spezifische Attribut übermittelt wird. Die letztere Form der Übermittlung von Daten ist daher selektiver und reduziert die Last des Netzwerkes [89]. Die Netzwerkkommunikation erfolgt in der HLA via TCP/IP oder UDP/IP [38]. Nicht erfolgreich zugestellte Datenpakete werden so lange erneut versandt, bis ihre Ankunft bestätigt wurde.

Die wichtigste Regel der HLA besteht im zentralen Datenaustausch über die RTI. Hieraus ergibt sich der Vorteil, dass keine simulationsbezogene Kommunikation erfolgt, die nicht vorher von der RTI überprüft wurde. Zudem behält die RTI stets die Kontrolle über die Zeitentwicklung in der Federation. Zeitkritische Datenübertragungen werden mit Zeitmarken versehen, deren korrekte Übermittlungsreihenfolge die RTI garantiert. Die Kausalität der Simulation wird somit zu jedem Zeitpunkt gewährt [89], [160]. Gleichzeitig stellt die zentrale Datenübermittlung der RTI einen Flaschenhals dar, was eine Erhöhung der Simulationszeit bei hohem Datenaufkommen zur Folge hat. Weitere Informationen zur verteilten Simulation mit der HLA sind in den folgenden Quellen [89], [38] und [155] dokumentiert.

Die Entwicklung des Brennstoffzellensystems erfolgte an drei Standorten des Unternehmens EADS Deutschland GmbH. Die Modellierung des Kondensators wurde in Ottobrunn durchgeführt, während die restlichen Systemkomponenten in Hamburg modelliert wurden. Zusätzlich wurde eine reale Brennstoffzelle eingesetzt, die mittels HIL in die verteilte Simulation eingebunden wurde. Durch den Einsatz der verteilten Simulation war es möglich, Modelle permanent an das fachspezifische Wissen der jeweiligen Standorte zu binden, um so zeitnah auf Modellierungsfehler oder Anpassungen der Modelle, bedingt durch die kontinuierliche Weiterentwicklung des Systems, reagieren zu können, ohne die Modelle zunächst zu übermitteln und anschließend in die Simulation zu integrieren. Zu Beginn des Projektes wurden einmalig die Schnittstellenparameter definiert, die zum Datenaustausch zwischen den Teilsystemen benötigt wurden.

Ein Vorteil der HLA besteht darin, dass die Schnittstelle zwischen dem einzelnen Modell und dem Netzwerk genau definiert ist. Zudem gewährleisten die Methoden der HLA die Netzwerkkommunikation sowie die Vermeidung von Inkonsistenzen [38]. Der Ablauf der Simulation ist abhängig von den Simulationsteilnehmern. Sind alle Teilnehmer ausschließlich in Software nachgebildet, besteht keine Veranlassung, die Simulation in Echtzeit ablaufen zu lassen. Ein komplett ereignisbasierter Ablauf kommt allerdings nicht infrage, da vorhandene

5.2 Realisierung des Simulationskonzepts

Regelkreise eine Zeitkomponente zur Steuerung benötigen. Bei der Verwendung von HIL ist die Simulationszeit in Einklang mit der realen Zeit (engl.: wall clock time) zu bringen [53]. Hierfür wurde die Federatezeit eines Federates mit der Uhr des Hostrechners synchronisiert. Das Zeitmanagement der HLA bietet z. B. die Möglichkeit den Zeitfortschritt in der Federation so lange zu stoppen, bis alle Federates einem Fortschritt zugestimmt haben.

Als Simulationsprogramm wurde Matlab/Simulink verwendet. Der Programmcode wurde in C++ entwickelt und die Brennstoffzelle wurde über das miniLAB™ 1008 der Measurement Computing Corporation angesteuert. Das Brennstoffzellensystem besteht neben der Brennstoffzelle aus einem Kondensator zur Wasserrückgewinnung und einer variablen Last. In Abbildung 5-29 sind die einzelnen Simulationsteilnehmer sowie der Datenaustausch zwischen den Federates dargestellt. Die verteilte Simulation besteht aus vier Simulationsteilnehmern und einem Hostrechner, auf dem die RTI ausgeführt wird. Zusätzlich zu den drei Teilsystemen des Brennstoffzellensystems wurde ein Monitor als vierter Simulationsteilnehmer eingeführt, um den gesamten Datenaustausch der verteilten Simulation zu kontrollieren.

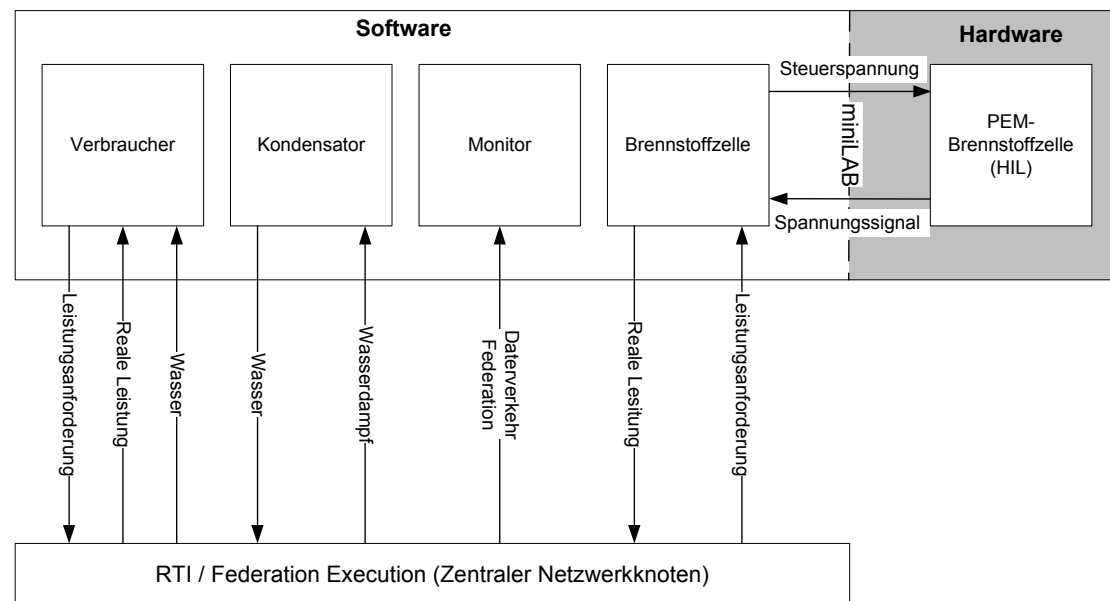


Abbildung 5-29 Verteilte Simulationsarchitektur des Brennstoffzellensystems

Basierend auf Daten aus einer Flugzeugverbraucherdatenbank, wird vom Simulationsteilnehmer „Verbraucher“ eine Leistungsanforderung an die Brennstoffzelle übermittelt. Für eine übersichtliche Darstellung der Systemparameter werden die von der Brennstoffzelle gelieferte Leistung sowie die Wasserrückgewinnung im Kondensator an diesen Federate übermittelt. Der Kondensator wurde mithilfe des Simulationsprogramms Matlab/Simulink modelliert und simuliert. Über eine C-Schnittstelle des Programms ist es möglich, den Kondensator in der verteilten Simulation zu integrieren und weiterhin in Matlab/Simulink zu simulieren. Aus der verteilten Simulation erhält er den Wasserdampfstrom der Brennstoffzelle und liefert als Ausgabe den auskondensierte Wasserfluss in Kilogramm pro Sekunde. Der Brennstoffzellen-Federate besteht aus der realen Brennstoffzelle und einem Softwareteil, der die Kommunikation mit der RTI übernimmt. Ausgehend vom Lastprofil des Verbrauchers, wird eine Leistungsanforderung an die Brennstoffzelle übermittelt. Diese Leistungsanforderung wird zunächst auf den Leistungsbereich der eingesetzten Brennstoffzelle skaliert. Die Schnittstelle zwischen der Hardware und der Software bildet das miniLAB 1008™. Das miniLAB™ ermöglicht eine Spannungssteuerung der an der Brennstoffzelle angeschlossenen regelbaren Last. In Abhängigkeit der Last ergibt sich für die Brennstoffzelle ein Betriebspunkt auf der Polarisationskurve (vgl. Abbildung 5-2) und eine dementsprechende Leistung. Mithilfe des miniLAB™ wird die Spannung über der Brennstoffzellenlast gemessen und an den Softwareteil des Brennstoffzellen-Federates zurückgegeben. Der Zugriff auf das Brennstoffzellenlabor erfolgt über die dem miniLAB™ beiliegende C++-Bibliothek, die in den Programmcode integriert werden muss. Über eine Stromspannungsfunktion, die vom Hersteller des miniLAB™ angegeben wird, werden Strom und Leistung der Brennstoffzelle berechnet. In Abhängigkeit vom Strom wird unter Zuhilfenahme der Gleichung (5-4) der erforderliche Was-

Gesamtkonzeptevaluierung

serstoffbedarf berechnet. Über die stöchiometrischen Verhältnisse aus Gleichung (5-1) und der Wasserstoffumwandlung in der Brennstoffzelle wird der produzierte Wasserdampf errechnet, der anschließend in dem Kondensator verflüssigt wird. Es wird deutlich, dass der gesamte Datenaustausch über die RTI verläuft. Um die Auslastung der RTI zu reduzieren, wurde aus diesem Grund der Datentransfer in der Federation mithilfe der Attribute realisiert.

Die einzelnen Federates der verteilten Simulation weisen einen nahezu identischen Aufbau auf. Sie unterscheiden sich im Programmcode lediglich durch ihre spezifischen Funktionen. Um den Eindruck des einheitlichen Grundaufbaus zu verstärken, wurde die Benutzeroberfläche zusätzlich einheitlich gestaltet (siehe Abbildung 5-30). Nach der Einarbeitung in die Bedienung eines Federates ist es möglich, die weiteren Federates intuitiv zu benutzen. Aus Gründen der Bedienerfreundlichkeit sind immer nur die Felder aktiv, deren Verwendung zum jeweiligen Zeitpunkt benötigt wird. Die einheitliche Vorgehensweise in der Programmierung sowie in der grafischen Darstellung der Federates führt nicht nur dazu, dass der Umgang für den Nutzer vereinfacht wird, zusätzlich verringert sich der Programmieraufwand, und die Wieder- bzw. Weiterverwendung der einzelnen Simulationsteilnehmer wird erhöht.

Die Federates wurden so programmiert, dass sie mehrere Aufgaben parallel abarbeiten können (engl.: multithreading). Diese Eigenschaft ermöglicht z. B. die gleichzeitige Bedienung der Oberfläche bei aktivem Datenaustausch. Die Federates bestehen jeweils aus drei Klassen. Die Windows-Forms-Klasse beinhaltet die Methoden, welche die grafische Benutzerschnittstelle beschreiben. In der Funktionsklasse ist die eigentliche Funktionalität der einzelnen Federates hinterlegt. Im Falle der Brennstoffzelle sind hier die Schnittstellen zum miniLAB™ sowie die Umrechnungsformeln hinterlegt. In einer dritten Klasse, der Federate-Ambassador-Klasse, wird der von der RTI ausgehende Datentransfer abgehört, und die für den jeweiligen Federate benötigten Daten werden an die beiden ersten Klassen weitergeleitet. Eine weitere Klasse wird von der RTI-Installation auf dem jeweiligen Federate-Host zur Verfügung gestellt. Diese Klasse beinhaltet Methoden zur Übermittlung von Daten an die RTI und ist für alle Federates identisch.

In Abbildung 5-30 ist die grafische Benutzeroberfläche der Simulationsteilnehmer exemplarisch am Beispiel des Verbraucher-Federates (engl.: load) dargestellt. Lediglich der Monitor-Federate weicht von dieser Darstellungsform ab, da er keine Funktionsmanagementzone und keine Variablenanzeige benötigt.

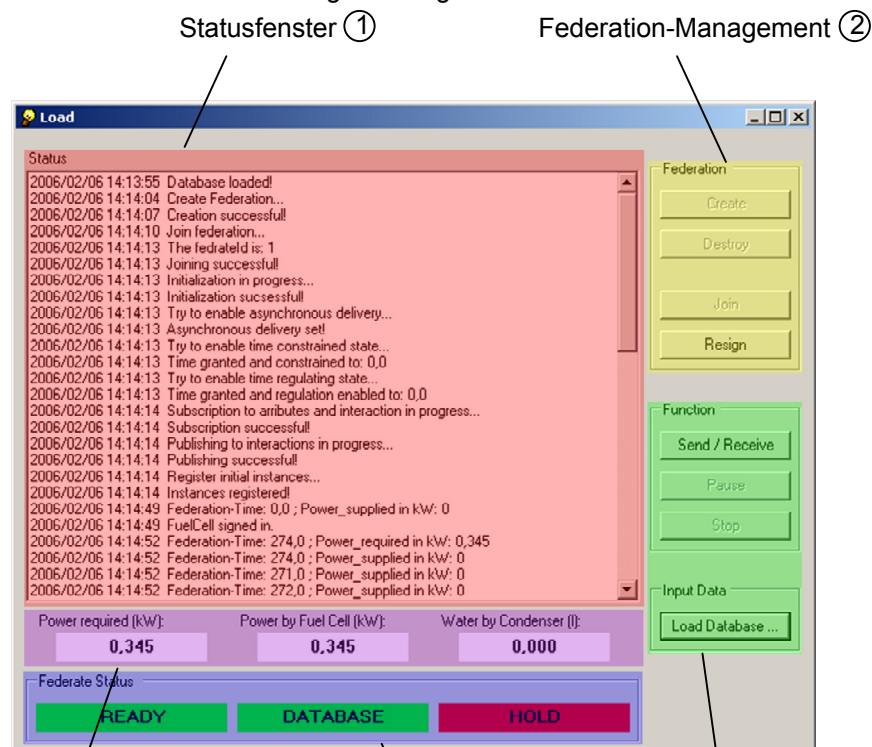


Abbildung 5-30 Grafische Benutzerschnittstelle des Verbraucher-Federates

5.2 Realisierung des Simulationskonzepts

Im Statusfenster (1) werden Fehlermeldungen oder Bestätigungen ausgeführter Kommandos des Federates angezeigt. Aus Gründen der Nachvollziehbarkeit werden die dargestellten Informationen mit einem Zeitstempel versehen. Die Statusanzeige ermöglicht dem Nutzer die Überwachung der verteilten Simulation. Es kann z. B. erkannt werden, welche Simulationsteilnehmer der Federation beigetreten sind (z. B. „FuelCell signed in“; siehe Abbildung 5-30) und wann die Simulation gestartet wurde. Zusätzlich wird der für den jeweiligen Federate benötigte Datentransfer angezeigt. Im Falle des Verbrauchers werden die Leistungsanforderung (Power_required) sowie die von der Brennstoffzelle gelieferte Leistung (Power_supplied) ausgegeben. Das Federation-Management (2) beinhaltet Funktionen, die zur Durchführung einer HLA-basierten verteilten Simulation benötigt werden:

- Gründen einer Federation
- Beitreten zu einer existierenden Federation
- Initialisieren von Objekten, Attributen, Interaktionsklassen und Parametern
- Erklärung über die Bereitschaft zum Veröffentlichen (engl.: publish) und Empfangen (engl.: subscribe) von Attributen und Interaktionen
- Zeitmanagementfunktionen
- Registrierung gegenüber der RTI
- Verlassen der Federation
- Beenden der Federation, sofern es das letzte verbleibende Federate ist

Diese Methoden sind in ihrem Aufbau in allen Federates identisch. Sie unterscheiden sich nur in der Anzahl und Zugehörigkeit der zu veröffentlichenden oder zu beziehenden Attribute und Parameter. Jedes, auch zukünftige, Federate sollte mit den Funktionen ausgestattet werden. Es wird bewusst das Wort „muss“ vermieden, weil es nicht zwingend erforderlich ist, dass jedes Federate die Federation erzeugen und auch wieder zerstören kann. Für einen selektiven Betrieb hat diese Eigenschaft den Vorteil, dass alle Teilnehmer nach Programmstart die Federation gründen und ihr beitreten können, ohne auf ein für diese Aufgabe zuständiges Federate zu warten. Diese grundlegenden für die verteilte Simulation mit der HLA benötigten Methoden erstrecken sich bei dem derzeitigen Simulationsaufbau über ca. 70 % des Quellcodes. Daraus resultiert, dass die Anzahl der Codezeilen der Federates, trotz unterschiedlichen Funktionsumfangs, nur in einem kleinen Bereich schwankt. Für weitere Informationen zu den Funktionen und Methoden der HLA sei an dieser Stelle auf folgende Literatur verwiesen [53], [89], [38].

In der Funktionsmanagementzone (3) befinden sich individuelle Einstellungsoptionen, bezogen auf die Funktion des Federates. Die Federate-Statusanzeige steht im direkten Zusammenhang zum Funktions- und Federation-Management. Durch die Verwendung einer Farbcodierung – rot: negativer Status und grün: positiver Status – wird dem Nutzer die Überwachung des Federates vereinfacht. In der Variablenanzeige werden die aktuellen Werte der Attribute, die ein Federate erhält oder versendet, dargestellt. Die Darstellung in der Variablenanzeige erfolgt stets zeitnah zum Empfang oder Versand neuer Werte. Im Falle des Verbraucher-Federates werden hier die Leistungsanforderung, die gelieferte Leistung der Brennstoffzelle und die Wasserrückgewinnung im Kondensator dargestellt.

5.2.4.1 Nutzung der HLA im Entwicklungsprozess

Die verteilte Simulation stellt dem Entwicklungsingenieur ein Werkzeug zur Verfügung, mit dem abteilungs- und standortübergreifende Simulationen möglich sind. Die benötigten Ressourcen bleiben in den lokalen Entwicklungsbereichen und können auch während der Simulation voll genutzt werden. Damit entfällt der Integrationsaufwand für die Zusammenführung von Teilmodellen an einem zentralen Standort. Treten durch fehlerhafte Modelle Probleme bei der Simulation auf, ist aufgrund des fachspezifischen Wissens am jeweiligen Standort eine zeitnahe Fehlerbehebung möglich. Die Simulationsteilnehmer bleiben permanent in erster Hand, für die beteiligten Entwickler anderer Teilsysteme sind lediglich die Schnittstellen zu den einzelnen Teilnehmern von Interesse. Als Folge dessen verkürzt sich zum einen die Entwicklungszeit, zum anderen werden Ressourcen auf das jeweilige Teilsystem konzentriert. Ein weiterer Vorteil besteht darin, dass Hardwarekomponenten unabhängig vom Standort in die Simulation integriert werden können. Mit der HLA ist es möglich, beliebig viele Hardware- und Softwarekomponenten in einer verteilten Simulation zusammenzuschließen. Während Einzelplatzsimulationen aufgrund der begrenzten Anzahl von Erweiterungssteckplätzen die Einbindung von HIL begrenzen, können mit verteilten Ressourcen die

Anschlüsse beliebig durch Hinzufügen weiterer Rechner skaliert werden. Für jede Hardwarekomponente kann ein eigener dedizierter Rechner eingesetzt werden, wodurch Ressourcenkonflikte vermeidbar werden.

Die HLA erfordert andere Herangehensweisen bei der Erstellung von Simulationen für technische Systeme als bei konventionellen Einzelplatzsimulationen. Komplexe Systemstrukturen sind in logische Teilprozesse zu zerlegen und in einer verteilten Simulationsarchitektur zu simulieren. Die Berechnungsgeschwindigkeit der verwendeten Hardware stellt somit keinen limitierenden Faktor mehr dar. Ist ein Simulationsmodell zu komplex, wird es so lange in logische Teilmodelle zerlegt, bis die vorhandene Rechenleistung ausreicht. Mit der vorangegangenen verteilten Brennstoffzellensystemsimulation wurden sowohl lokal als auch verteilt Berechnungen durchgeführt. Die lokale Simulation lief in 13-facher Echtzeit, während die verteilte Simulation in 27-facher Echtzeit ablief und damit mehr als doppelt so schnell war.

Aufgrund des einheitlichen Aufbaus der Federates konnte der Programmieraufwand reduziert werden. Ein Großteil des Codes wird für die Methoden der HLA benötigt und ist in allen Federates identisch. Sie unterscheiden sich lediglich im Datenaustausch und in der Funktionalität, sodass ein Großteil des Programmcodes wiederverwendet werden konnte. Zusätzlich können vorhandene Federates in beliebiger Anzahl in weiteren HLA-basierten Federationen eingesetzt werden. Ein weiterer Vorteil der HLA besteht darin, dass sich der Komplexitätsgrad von Teilmodellen stufenlos anpassen lässt. Steigt die Modelldetaillierung mit voranschreitender Entwicklungszeit an, bleiben die Schnittstellen davon unberührt. Voraussetzung hierfür ist, dass die Schnittstellen des neuen Modells mit denen des alten Modells übereinstimmen. Eine steigende Detaillierung resultiert in einer Erhöhung des Rechenaufwands, sodass es zu weiteren Aufteilungen in logische Teilsysteme kommen kann. Kommt ein weiteres Federate hinzu, muss dieses bei der Gründung der Federation bekannt sein und von der Federation Execution (vgl. Abbildung 5-28) initialisiert werden. Weiterhin ist zu analysieren, welche Federates direkten Kontakt mit dem hinzugekommenen Teilnehmer haben. Diese Federates sind um den Zugriff auf die neue Objektklasse zu erweitern. Durch die C++- und Java-Programmierschnittstelle der RTI ist es zudem möglich, beliebige Simulationsplattformen in die Simulation einzubinden. Sofern ein Simulationsprogramm über eine dieser Schnittstellen verfügt, kann es problemlos in die HLA-Umgebung integriert werden. Dabei bleiben die Vorteile der programmeigenen Lösungsalgorithmen erhalten. Somit ist es möglich, immer das am besten geeignete Simulationsprogramm für die Berechnung der verschiedenen Teilsysteme einzusetzen.

Der Einsatz einer HLA-basierten verteilten Simulation ist aber auch mit Nachteilen verbunden. Die Einarbeitungszeit in die Programmierung der Federates sowie die Planung der Federation sind zeitintensiv und binden dementsprechend viele Ressourcen. Vor Beginn der Programmierung müssen alle Schnittstellen zwischen den Simulationsteilnehmern spezifiziert werden. Zudem müssen für eine erfolgreiche Durchführung der Simulation aufwendige Zeitsynchronisationsverfahren, wie in [53] beschrieben, eingesetzt werden, und somit ist der Einsatz professioneller Programmierer unausweichlich. Allerdings relativiert sich der hohe Implementierungsaufwand durch die hohe Wiederverwendbarkeit der einzelnen Simulationsteilnehmer wie auch der gesamten Federation.

Zusammenfassend lässt sich sagen, dass die verteilte Simulation ein effizientes Werkzeug für die Simulationskopplung im Entwicklungsprozess darstellt. Bedingt durch den hohen Implementierungsaufwand ist von einem kontinuierlichen Einsatz allerdings abzuraten. Für die punktuelle Verwendung während der systemübergreifenden Simulationen zum Ende der Konzept- und Detaillierungsphase (M5 und M7; vgl. Abbildung 3-1) stellt sie eine Möglichkeit zur Analyse der Systeminteraktionen dar. Zusätzlich bietet sie eine Plattform für die gemeinsame Simulation von Hard- und Software während der Testphase.

5.3 Informationsmanagement

Für den kontinuierlichen Einsatz von Simulationstechniken in der Entwicklung von Flugzeugsystemen spielt die Informationsbereitstellung eine wichtige Rolle. Aus diesem Grund wurde in Kapitel 4.2.2 ein zentrales Informationsmanagement für die Bereitstellung von Simulationsdaten sowie eine theoretische Realisierung der zentralen Informationsbereitstellung (siehe Abbildung 4-16) beschrieben. Im Folgenden werden die standardisierte Bereitstellung von Simulationsdaten sowie der nutzerbezogene Zugriff auf die in der Informationsbasis enthaltenen Daten dargestellt.

5.3.1 Standardisierte Bereitstellung von Simulationsdaten

Um die Vorteile der computergestützten Simulation ausschöpfen zu können, müssen Simulationsergebnisse für den Anwender übersichtlich und für alle Simulationsprogramme kompatibel zur Verfügung stehen. Für den Austausch von Simulationsdaten existiert bereits eine Vielzahl an kommerziellen Softwarelösungen, wie z. B. der SimManager des Unternehmens MSC Software, Versionsverwaltungssysteme oder eRoom von EMC Documentum. Für weitere Informationen sind im Anhang ein Überblick über diese Programme sowie eine Beschreibung der individuellen Vor- und Nachteile enthalten (siehe Kapitel 7.4). Alle diese Softwarelösungen weisen einen gemeinsamen Nachteil auf: Es ist nicht möglich, Simulationsdaten automatisch in ein kompatibles Format des vom Anwender benutzten Simulationsprogramms zu konvertieren. Das Konvertieren wird zurzeit ausschließlich den eingebetteten Funktionen der Simulationsprogramme überlassen. Dabei tritt das Problem auf, dass jedes Programm die Konvertierung in jedes Format beherrschen muss. Die Anzahl der Konvertierungsroutinen steigt damit quadratisch mit der Anzahl der Simulationsprogramme. Durch die Verwendung eines zentralen Austauschformats steigt der Aufwand nur noch linear mit der Anzahl der Simulationsprogramme, da in jedem Programm nur noch die Konvertierung in das Standardformat implementiert werden muss. Zusätzlich zur Aufwandsreduzierung werden mögliche Fehlerquellen beim Austausch entfernt. Ein weiteres Defizit besteht darin, dass bisher zwei Programme zum Durchführen der Simulationen und zum Ablegen der Simulationsdaten in einem zentralen Datenspeicher eingesetzt werden.

Für den Austausch zwischen den Simulationsprogrammen wird ein geeignetes Datenformat benötigt. Wie schon bei den Softwareprogrammen existieren verschiedene Datenformate, wie z. B. das Hierarchical Data Format (Abk. HDF), ein wenig speicherintensives und flexibles Binärdatenformat, Comma Separated Values (Abk. CSV; deutsch: durch Kommas getrennte Werte) und die Extensible Markup Language (Abk. XML; deutsch: erweiterbare Auszeichnungssprache). Das HDF ist plattformabhängig und aus diesem Grund nicht als universelles Simulationsdatenaustauschformat einsetzbar. Das CSV-Format ist auf Simulationsdaten beschränkt, die sich genau in einer Matrix beschreiben lassen. Diese Nachteile weist das XML-Format nicht auf, es ermöglicht die Beschreibung beliebiger Daten. Der Vorteil von XML gegenüber anderen Austauschformaten ist, dass entsprechende Syntaxanalysatoren (engl.: parser) in allen gängigen Programmiersprachen verfügbar sind und dass die Definition der Sprache von einem unabhängigen Gremium durchgeführt wird. Eine XML-Programmierschnittstelle (Abk. API; engl.: Application Programming Interface) enthält außer dem Parser auch weitere Algorithmen zum Validieren von Inhalten (XML-Schemas und Dokumenttyp-Definition) und zur Transformation der Inhalte in andere Formate (Extensible Stylesheet Language). Der Nachteil von XML besteht darin, dass Daten im XML-Format deutlich mehr Speicher belegen, als es in den meisten anderen Formaten der Fall ist. Weitere Informationen zu den vorgestellten Austauschformaten sind im Anhang enthalten (siehe Kapitel 7.4.2). Die XML kommt bereits als Austauschformat für Simulationsdaten zum Einsatz. Olbert benutzt das XML-Format, um Simulationsdaten gestaltorientierter Simulationen standardisiert abzulegen und die Ergebnisse anschließend über einen Webbrowser anzuzeigen. Im Unternehmen Boeing wird ein XML-basiertes Datenformat (Abk. SRML, engl.: Simulation Reference Markup Language) für den Austausch und die Integration von Simulationsmodellen eingesetzt. Es ist sogar möglich, in SRML geschriebene Modelle auszuführen [14], [135]. Aus diesen Gründen wird im Folgenden XML für den standardisierten Austausch von Simulationsdaten verwendet.

Ein sequenzieller Austausch von Simulationsdaten zwischen Simulatoren wird als lose Simulationskopplung bezeichnet (siehe Abbildung 2-9). Sie kann immer dann eingesetzt werden, wenn keine signifikanten Rückwirkungseinflüsse vorhanden sind oder der Arbeitspunkt vor der Simulation feststeht. Bei losen Kopplungen tritt das Problem auf, dass die Simulatoren proprietäre Formate verwenden. Zur Durchführung der gesamten Simulation muss ein Konverter existieren. Aus diesem Grund wird in diesem Kapitel die Umsetzung einer standardisierten Bereitstellung von Simulationsdaten beschrieben. In einer Software wird der Vorgang der Simulationsdatenkonvertierung und -ablage automatisiert und damit die Effizienz des Austauschs von Simulationsdaten gesteigert. Es wird ein verteiltes System beschrieben, das in der Lage ist, von verschiedenen Computersystemen Simulationsdaten an einen Server zu übertragen und von dem Server abzurufen. Das verteilte System zeichnet sich durch folgende Eigenschaften aus:

- Die Bedienung erfolgt über eine grafische Benutzeroberfläche

Gesamtkonzeptevaluierung

- Austauschbarkeit des Datenbanksystems (Abk. DBMS; z. B. PDM)
- Simulatoren und Datenbanken sind über ein Netzwerk verbunden
- Automatische Konvertierung in ein XML-Standardformat
- XML-Proprietären für die Simulationsdaten
- Definition eines einfachen Protokolls zur Anbindung von Simulatoren an den Server
- Generische Datenbankanbindung
- Verwaltung des Servers über eine Webschnittstelle
- Erweiterbarkeit der Anwendung
- Übersichtliche Darstellung benutzerspezifischer Daten durch individuelle Zugriffsbe-
rechtigung
- Automatische Validierung der Daten vor dem Ablegen in der Datenbank

In Abbildung 5-31 ist der Aufbau des verteilten Systems dargestellt. Nach Beendigung einer Simulation werden die Simulationsergebnisse über ein Netzwerk an einen Server versendet und in einer Datenbank abgelegt. Der Server wird von einem unabhängigen Rechner im Netzwerk über einen Webbrowser verwaltet.

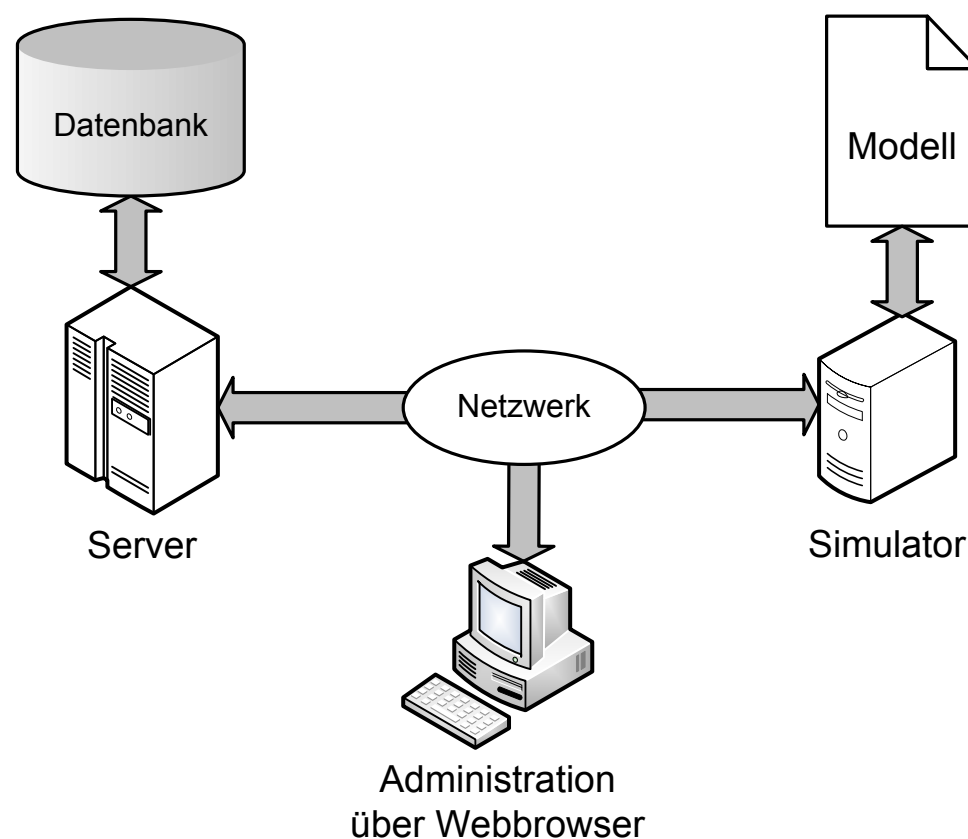


Abbildung 5-31 Aufbau des verteilten Systems

Für die Softwareumsetzung des verteilten Systems wurde die objektorientierte Programmiersprache Java eingesetzt. Java bietet mit der Java Database Connectivity (Abk. JDBC) eine DBMS-unabhängige Datenbankverbindung und ermöglicht somit die Erweiterbarkeit und Austauschbarkeit verwendeter Datenbankmanagementsysteme. Zudem verfügt die Programmiersprache über eine Netzwerk-API für die Kommunikation zwischen Client, Server und Browser. Java-Programme sind plattformunabhängig, sodass ein Betriebssystemwechsel keine Neuentwicklung oder Portierung zur Folge hat. Hierin besteht allerdings auch ein Nachteil: Java-Programme sind durch die Ausführung in einer virtuellen Maschine systembedingt nicht performant, müssen dafür aber nicht für jede Plattform neu kompiliert werden [1]. Die Performance spielt bei dieser Problemstellung eine untergeordnete Rolle, daher ist diese negative Eigenschaft von Java zu vernachlässigen.

Aus Kostengründen wurde die frei erhältliche Datenbanksoftware MySQL für das Datenbankmanagementsystem eingesetzt. MySQL verfügt über Klasse-4-JDBC-Treiber und ermöglicht somit eine unkomplizierte und datendurchsatzstarke Einbindung in Java. Aufgrund

5.3 Informationsmanagement

der mangelhaften Speicherverwaltung sollte MySQL in einer kommerziellen Lösung nicht eingesetzt werden. In der Regel treten diese Probleme erst bei großen Datenmengen auf. Da bei der Umsetzung der standardisierten Bereitstellung von Simulationsdaten im Folgenden nur eine geringe Datenmenge ausgetauscht wird, ist dieses Problem zu vernachlässigen. Weitere Datenbanksysteme sind z. B. Oracle, Sybase, Postgress, DB2 von IBM und MySQL von Microsoft. Für die individuelle Auswahl einer geeigneten Datenbanksoftware stellt das Bundesamt für Sicherheit in der Informationstechnik eine Vorgehensweise im Internet zur Verfügung [22].

5.3.1.1 Umsetzung des verteilten Systems für den standardisierten Datenaustausch

Im Folgenden wird der standardisierte Datenaustausch zwischen den zwei Simulationsprogrammen Matlab und Fluent beschrieben. Eine Schnittstellenspezifikation des verteilten Systems ermöglicht allerdings die Einbindung beliebiger Simulationsprogramme. In allen Fällen werden Simulationsdaten aus einem proprietären Datenformat in ein standardisiertes XML-Format konvertiert und in einem beliebigen Datenbankmanagementsystem abgelegt. Die XML-Dateien stehen damit zur Weiterverwendung in den angebundenen Simulationsprogrammen zur Verfügung. Das verteilte System besteht aus vier Anwendungsklassen (Simulator, Transaktionsserver, Verwaltung und Datenbank). Die Abbildung 5-32 zeigt den Datenfluss zwischen diesen Anwendungsklassen.

Die Datenübertragung erfolgt über einen zentralen Transaktionsserver, er bildet das Koppellement zwischen der Datenbank und allen anderen Komponenten des verteilten Systems. Aus diesem Grund werden die Zugriffsbeschränkungen auf die Datenbanken an dieser Stelle definiert. Die Anwendungsklasse der Simulationsprogramme, die Daten an den Server sendet und abrufen, besteht aus zwei Anwendungen. Zum einen existiert ein Programm zum Einlesen, Umwandeln und Übermitteln von Dateien an den Server, zum anderen gibt es eine integrierte Anbindung an Matlab. Damit ist es möglich, Workspace-Daten direkt aus Matlab in einer Datenbank einzutragen oder auszulesen. Dieser Datenbankzugang automatisiert den Arbeitsablauf des Auslesens aus der Datenbank und spart dadurch bei jeder Transaktion Arbeitszeit. Pro Transaktionsserver können mehrere dieser Anwendungen parallel eine Verbindung herstellen. Für die Verwaltung des Servers existiert ein Java-Applet (ein Programm, das im Rahmen eines anderen Programms betrieben wird), das über einen Browser angezeigt wird. Nach dem Anmeldevorgang kommuniziert das Applet direkt mit dem Transaktionsserver ohne den Umweg über den Browser. So kann gewährleistet werden, dass sich nur Benutzer mit Verwaltungsrechten an dem Server anmelden können. Die Simulations- und Verwaltungsdaten werden in einem DBMS eingestellt. Die Verwendung des generischen JDBC-Treibers ermöglicht die Einbindung beliebiger DBMS, ohne dass die Schnittstelle erneut programmiert werden muss.

Die von den Simulationsprogrammen übermittelten proprietären Daten werden vor dem Ablegen im DBMS anhand einer XML-Schema-Definition (Abk. XSD) validiert. Für den Austausch von Simulationsdaten sind verschiedene Variablentypen in der XSD-Datei hinterlegt. Zudem wurde in der Schemadefinition berücksichtigt, dass Simulationsdaten in Form von skalaren Größen, Vektoren, Matrizen oder Strukturen auftreten können. Stimmt der Aufbau einer vom Simulationsprogramm stammenden XML-Datei nicht mit der Schemadefinition überein, wird diese nicht in der Datenbank abgelegt. So wird gewährleistet, dass nur valide Simulationsdaten in der Datenbank enthalten sind und keine Konvertierungsfehler beim Auslesen auftreten können.

Als Protokoll für die Kommunikation zwischen den Instanzen des verteilten Systems wird das Hypertext-Übertragungsprotokoll (Abk. HTTP; engl.: Hypertext Transfer Protocol) in der Version 1.0 verwendet [10]. Dieses Protokoll ist einfach zu implementieren und zu erweitern. Es bietet die Möglichkeit, beliebige Daten zu transportieren und im Kopf eines Pakets beliebig viele Metadaten unterzubringen. Es sind nur die Methoden PUT und GET implementiert. Jede andere Methode führt zum Antwortcode „501 Not Implemented“. PUT ist für das Übertragen von Informationen zuständig, während GET Informationen anfordert. Die Basis des Übertragungsprotokolls bilden bidirektionale Softwareschnittstellen eines Übertragungssteuerungsprotokolls, sogenannte TCP-Sockets (engl.: Transmission Control Protocol), die die verbindungsorientierte Kommunikation sicherstellen (OSI-Modell: Transportlayer [Schicht 4]).

Gesamtkonzeptevaluierung

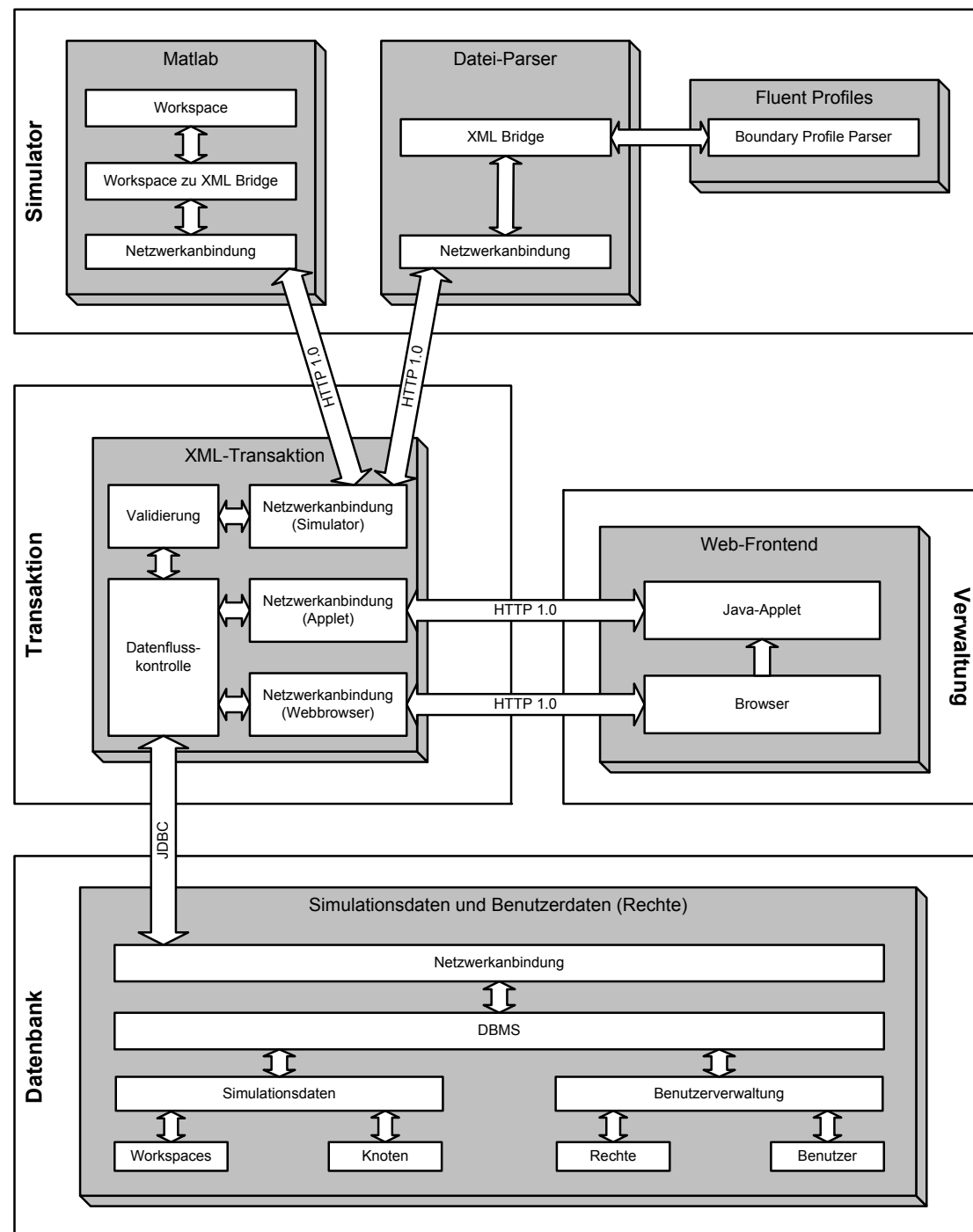


Abbildung 5-32 Topologie des Datenaustauschsystems

Im folgenden Abschnitt wird das Datenbankmodell zur Speicherung der Simulationsdaten in einer Datenbank beschrieben. Die Beschreibung dieses Modells erfolgt mit der Terminologie des Gegenstands-Beziehungs-Modells (Abk. ERM; engl.: Entity Relationship Modell) [28]. Für das Datenbankmodell werden drei Entitäten (engl.: entities) verwendet. Für die Ablage der Simulationsdaten wird eine Entität benötigt, in der die Simulationsdaten mit ihren Metainformationen aufgenommen werden können (siehe Abbildung 5-33). Zu den Meta-informationen gehört eine Beschreibung („name“), in der mit wenigen Worten der Inhalt zusammengefasst wird. Darüber hinaus existiert eine Gültigkeitskennzeichnung („valid“), in der hinterlegt ist, ob die Daten validiert sind. Das Datum („datum“) sowie die eigentlichen Simulationsdaten werden ebenfalls in einzelnen Feldern gespeichert. Um die Simulationsdaten strukturiert in der Datenbank abzulegen, wird eine weitere Entität benötigt, die eine Struktur repräsentiert. Um einem Benutzer den Umgang zu erleichtern, wurde eine Struktur ausgewählt, die der Ordnerstruktur eines herkömmlichen Dateisystems entspricht: Ein Ordner

5.3 Informationsmanagement

enthält Ordner und Dateien. Analog dazu enthält in diesem Datenbankmodell ein Knoten weitere Knoten und Simulationsdaten (siehe Abbildung 5-33). Jedem Knoten wird ein Name („name“) zugeordnet, über die Eins-zu-n-Relation („struktur“) wird die Ordnerstruktur realisiert.

Für den individuellen Zugriff auf Simulationsdaten wird eine weitere Entität benötigt, in der Informationen über autorisierte Benutzer abgelegt sind. Ein Benutzer hat einen Anmeldenamen („username“), einen Benutzernamen („name“), Kontaktdaten („kontakt“) und ein Passwort („passwort“), mit dem er sich anmeldet. Um den Zugriff des Benutzers auf die Datenstruktur einzuschränken, wird eine m-zu-n-Relation benötigt, die einen Benutzer mit einem Knoten verbindet. In dieser Relation wird zusätzlich die Verknüpfungsart gespeichert. Die Verknüpfungsart zwischen Benutzern und Knoten wird im folgenden Recht genannt. Es gibt fünf verschiedene Rechte:

- Der Benutzer darf am Knoten nicht lesen.
- Der Benutzer darf an dem Knoten das, was er an dem Elternknoten darf (erben).
- Der Benutzer darf am Knoten lesen.
- Der Benutzer darf am Knoten schreiben.
- Der Benutzer darf den Knoten verwalten.

Es gibt zwei verschiedene Arten von Rechten. Zum einen die knotenspezifischen Rechte, die einem Benutzer an einem Knoten ein Recht zuweisen. Zum anderen die effektiven Rechte, die für jeden Knoten berechnet werden müssen. In die Berechnung eines effektiven Rechts fließen die knotenspezifischen Rechte der Elternknoten mit ein. Für die Berechnung der effektiven Rechte wurde ein regelbasierter Algorithmus entwickelt. Eine Regel lautet z. B., dass jeder Benutzer an einem Knoten genau ein Recht hat. Ist dieses Recht nicht explizit angegeben, so wird das Recht implizit vererbt. Wenn an einem Knoten das Recht „nicht lesen“ vergeben ist, dann darf dieser Knoten vom Benutzer nicht beschrieben oder verwaltet werden. Verwalten impliziert schreiben und schreiben impliziert lesen usw. Der Algorithmus ermöglicht damit eine automatische Rechtevergabe, ausgehend von explizit vergebenen knotenspezifischen Zugriffsrechten. Für den Benutzer des verteilten Systems hat die individuelle Zugriffsbeschränkung zur Folge, dass nur auf relevante Daten zugegriffen werden kann und damit eine Datenüberflutung vermieden wird.

Das Datenbankmodell enthält alle oben beschriebenen Entitäten und Relationen. Die Abbildung 5-33 zeigt das vollständige Entity-Relationship-Diagramm. In der Entität „inhalt“ befinden sich die Simulationsdaten mit Metainformationen. In ihr müssen zwei Fremdschlüssel (Verweis auf eine andere Entität) generiert werden. Ein Fremdschlüssel verweist auf den Benutzer, der den Eintrag erzeugt hat, der andere verweist auf den Knoten, an dem die Daten abgelegt sind. Die Relation „erstellt“ legt fest, dass ein Inhalt genau einen erzeugenden Benutzer hat. Die Relation „gespeichert“ beschreibt, dass an einem Knoten mehrere Simulationsdaten gespeichert sein können. Die Tabelle „knoten“ beinhaltet die Speicherorte, die in der Datenbank verfügbar sind. Ein Speicherort hat einen Namen und eine Referenz auf einen weiteren Speicherort, an dem sich der Knoten befindet. In der Tabelle „benutzer“ sind die Benutzerkonten hinterlegt. Der Anmelde-name „username“ muss eindeutig sein. Zwischen den Entitäten „benutzer“ und „knoten“ existiert die m-zu-n-Relation „rechte“, die festlegt, welcher Benutzer an welchem Knoten welche Rechte besitzt. Die Art des Rechts ist die Eigenschaft „ebene“ der Relation, sie repräsentiert damit eines der fünf Rechte.

Implementierung des Transaktionsservers

Der Transaktionsserver stellt die zentrale Komponente des verteilten Systems dar (siehe Abbildung 5-32). Zu seinen Aufgaben gehören die Authentifizierung und Autorisierung der Benutzer, das Validieren der Simulationsdaten, das strukturierte Ablegen der Simulationsdaten in der Datenbank sowie der Betrieb eines integrierten Webservers zur Serververwaltung. Der Transaktionsserver bietet seine Funktionalität mehreren Clients an. Aus diesem Grund müssen vom Transaktionsserver multiple Aufgaben gleichzeitig bearbeitet werden. Das sind z. B. das Akzeptieren von eingehenden Netzwerkverbindungen und das Bearbeiten von Anfragen der Clients. Damit der Server Anfragen von mehreren Clients gleichzeitig bearbeiten kann, wurden Threads verwendet, welche es erlauben, mehrere Aufgaben innerhalb eines Prozesses gleichzeitig zu bearbeiten. Zum Zeitpunkt des Programmstarts existiert lediglich der Hauptprogrammthread. Dieser startet drei weitere Verbindungsthreads, die auf eingehende Verbindungen warten. Wenn ein Client eine Verbindung mit dem Server herstellt, startet einer dieser drei Threads einen Kommunikationsthread, dessen Aufgabe darin besteht,

die Kommunikation mit diesem einen Client abzuwickeln. Damit bleiben die drei Verbindungsthreads für weitere Clients verfügbar, während jede neue Client-Server-Verbindung von einem eigenen Kommunikationsthread übernommen wird. Auf diese Weise können beliebig viele Kommunikationsthreads gleichzeitig ausgeführt werden. Mögliche Clients sind z. B. Webbrowser, Verwaltungsapplets oder Simulationsprogramme. Um beliebig viele dieser Threads verwalten zu können, existiert eine verkettete Liste, in der alle aktiven Kommunikationsthreads eingetragen sind. Über den Hauptprogrammthread werden die in der Liste eingetragenen Threads überwacht und nach Beendigung der Transaktion beendet. In Abbildung 5-34 ist die Organisation der Threads dargestellt. Aus Gründen der Übersichtlichkeit wurden nur zwei Verbindungsthreads dargestellt.

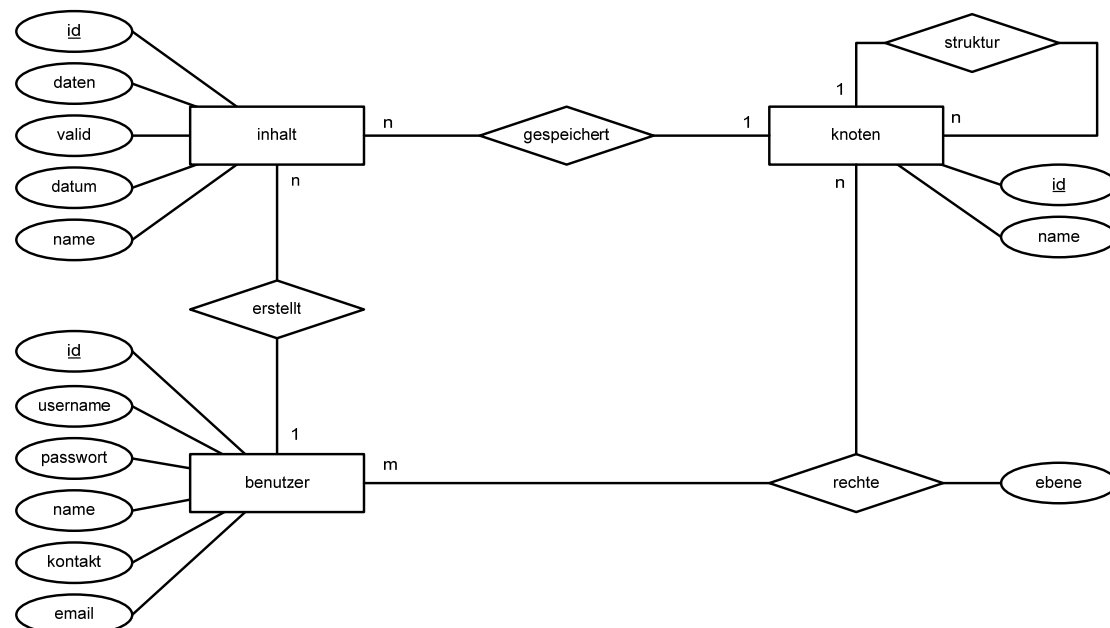


Abbildung 5-33 Entity-Relation-Diagramm der Simulationsdatenbank

Die Verbindung zwischen dem Server und den Clients basiert auf der HTTP-Protokollspezifikation. Im verteilten System sind bereits Matlab, ein Datei-Parser (Programm zur Umwandlung einer beliebigen Eingabe in ein anderes Format), und eine Webschnittstelle (engl.: Web-Frontend) als Clients implementiert (siehe Abbildung 5-32). Eine Bedingung für die Kommunikation zwischen Server und Client besteht darin, dass sich der Benutzer mit einem gültigen Benutzernamen und zugehörigen Passwort angemeldet hat. Diese Benutzerdaten müssen einmalig bei der Anmeldung angegeben werden, sie werden in der Kopfzeile des HTTP-Protokolls hinterlegt und bei sämtlichen Kommunikationen zwischen Server und Client übermittelt und überprüft. Nachdem die Netzwerkverbindung zwischen den Teilnehmern des verteilten Systems hergestellt wurde, muss der Server die Anfragen des Clients bearbeiten. Die Verbindung zur Datenbank wird über einen JCDB-Treiber realisiert, was die Einbindung beliebiger Datenbankmanagementsysteme möglich macht.

Es kann vorkommen, dass ein und dieselbe Abfrage mehrfach durchgeführt wird und jedes Mal denselben Rückgabewert annimmt. Für diesen Fall muss nicht immer erneut eine Datenbankabfrage durchgeführt werden. Aus diesem Grund ist es sinnvoll, das Ergebnis einer Abfrage zu speichern und bei erneutem Aufruf einfach aus dem Speicher zu laden. Für das temporäre Zwischenspeichern von Daten wurde ein SQL-Abfragen-Cache verwendet. Durch den Cache wird die Verbindung zwischen Transaktionsserver und DBMS entlastet.

Wie bereits beschrieben, erfolgt die Verwaltung des Servers über eine Webschnittstelle. Aus diesem Grund ist auf dem Transaktionsserver ein Webserver implementiert worden. Der Webserver stellt dem Webbrowser ein Java-Applet über einen integrierten Webserver zur Verfügung, das von diesem Browser angezeigt und vom Benutzer bedient wird. Ein Java-Applet hat gegenüber einem Web-Frontend den Vorteil, dass es auf Benutzereingaben schneller reagiert und daher von Anwendern eher als vollwertiges und effizientes Programm akzeptiert wird. Das ist die Grundvoraussetzung dafür, dass der Transaktionsserver in der Praxis von Anwendern regelmäßig genutzt wird. Die Eingabe von Benutzername und Passwort erfolgt im Java-Applet. In Abbildung 5-35 ist die Startseite des Webserver dargestellt.

5.3 Informationsmanagement

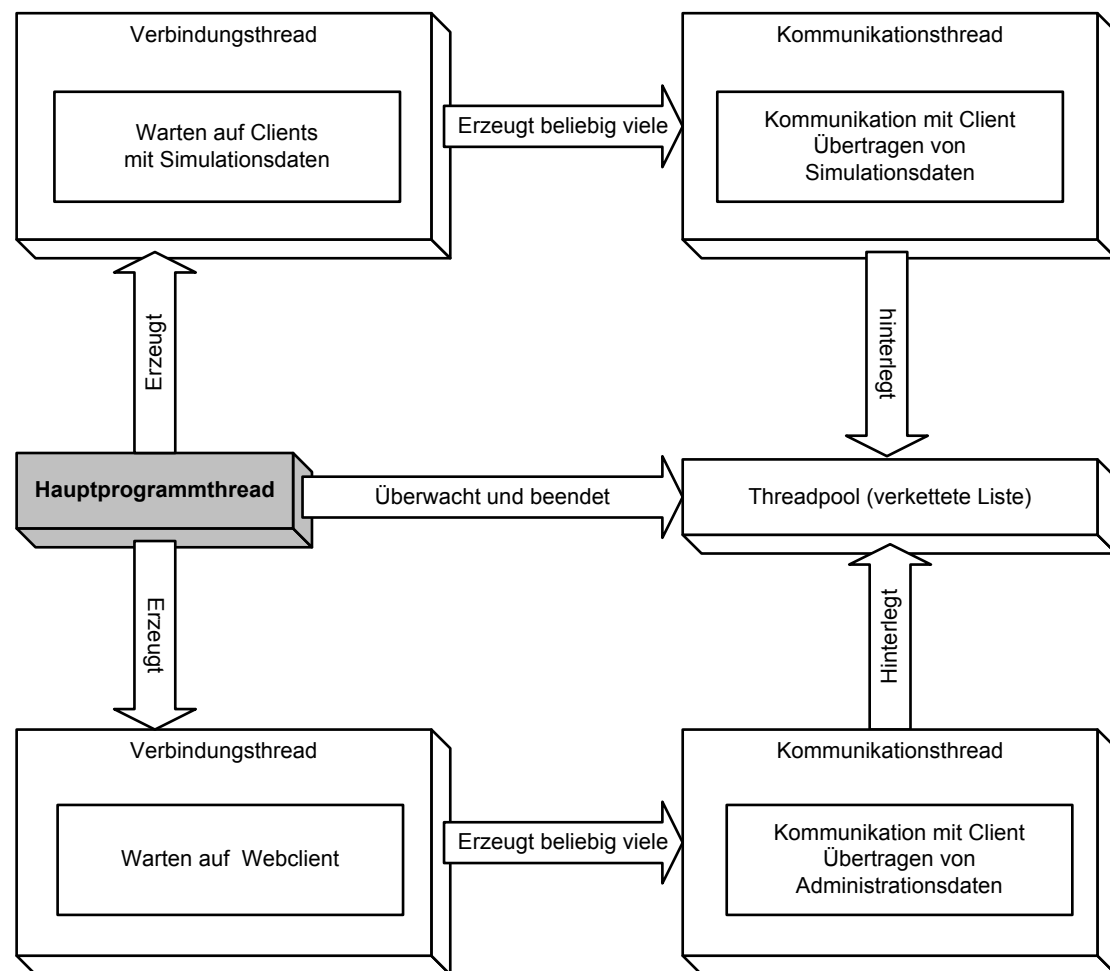


Abbildung 5-34 Organisation der Threads

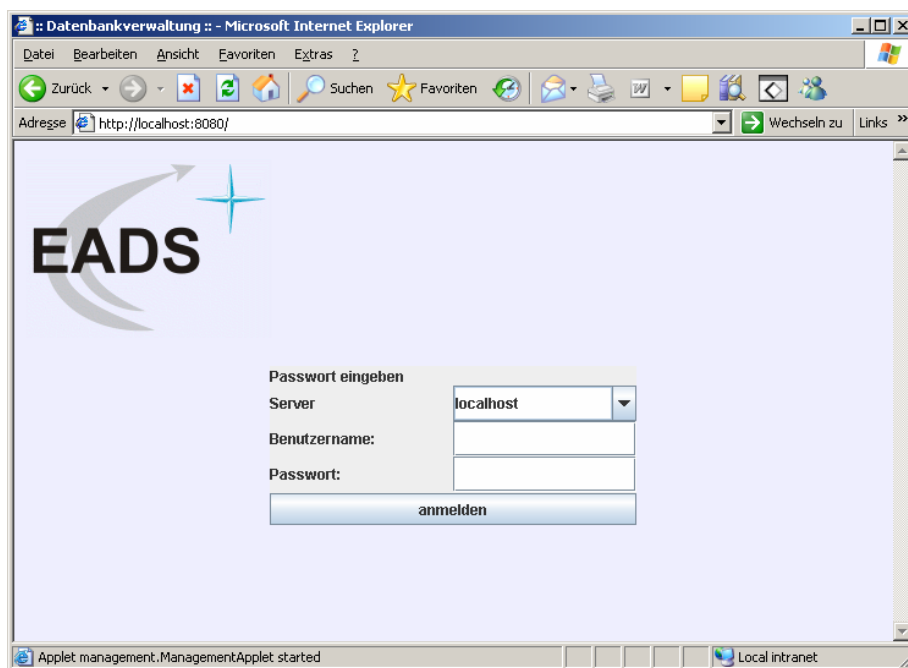


Abbildung 5-35 Startseite des Webservers

Anbindung der Simulationsprogramme

In diesem Abschnitt wird die Anbindung von Simulationsprogrammen an das verteilte System exemplarisch für Matlab und im Anschluss daran für Fluent beschrieben. Das mathematisch orientierte Simulationsprogramm Matlab rechnet intern mit Matrizen und bietet durch eine integrierte Scriptsprache die Möglichkeit, Abläufe zu automatisieren. Die Ausdrücke, die in dieser Scriptsprache verwendet werden, lassen sich zudem in einer Kommandozeile ausführen. In der Scriptsprache lassen sich auch Java-Ausdrücke mit einer in Matlab integrierten Java Virtual Machine (Abk. JVM) zur Ausführung des Java-Bytecodes auswerten. Die Simulationsergebnisse werden in Variablen im Arbeitsbereich (engl.: workspace) abgelegt [104]. Das Exportieren der Simulationsdaten aus dem Matlab Workspace ist in zwei Phasen unterteilt. In der ersten Phase wird der Workspace von einem Matlab-Script ausgelesen und in eine sogenannte Document-Object-Model (Abk. DOM) Baumstruktur konvertiert. Das Document Object Model ist eine Programmierschnittstelle für den Zugriff auf HTML- oder XML-Dokumente. Sobald der DOM-Baum fertiggestellt ist, wird dieser in ein XML-Dokument umgewandelt und in der zweiten Phase an eine Java-Klasse zur Ablage in der Datenbank übergeben. In der Java-Klasse erfolgt zunächst eine Authentifizierung des Benutzers. Sind Anmeldenamen und das Passwort für den Benutzer in der Datenbank hinterlegt, erscheint die in Abbildung 5-36 dargestellte grafische Benutzerschnittstelle. Der Benutzer wird aufgefordert, einen Namen der Simulationsdaten zu bestimmen und die zu speichernden Workspace-Variablen zu wählen. Das Kontrollkästchen in der oberen rechten Ecke legt zusätzliche Metainformation zum Datensatz fest.

Name	Typ	Speichern
ans	numeric	<input type="checkbox"/>
bar	char	<input checked="" type="checkbox"/>
foo	numeric	<input checked="" type="checkbox"/>
foobar	struct	<input checked="" type="checkbox"/>
test	numeric	<input checked="" type="checkbox"/>

Abbildung 5-36 Java-Applet zur Auswahl der Workspace-Variablen in Matlab

In Abhängigkeit von der individuellen Zugriffsberechtigung, die von einem Systemadministrator festgelegt wird, kann der Benutzer mit der Suchenfunktion Simulationsdaten an einem Speicherort in der Datenbank ablegen. Die verfügbaren Speicherorte werden in einem weiteren Dialogfenster angezeigt (siehe Abbildung 5-37). Für die Speicherung von Simulationsdaten in der Entwicklung von Flugzeugsystemen eignet sich die Strukturierung in ATA-Kapitel (vgl. Abbildung 4-18).

Wurden alle Eingaben korrekt durchgeführt, ist der Knopf zum Versenden des Workspaces aktiviert. Eine Betätigung des Knopfes führt zur Übermittlung der Simulationsdaten. Wie bereits beschrieben, überprüft der Transaktionsserver zunächst, ob der Benutzer für die Transaktion berechtigt ist und ob der XML-Workspace gemäß der Schemadefinition valid ist. Ein Fehler bei dieser Überprüfung führt zum Abbruch der Transaktion. Analog zum Datenexport erfolgt der Import ebenfalls in zwei Phasen.

In der ersten Phase wählt der Benutzer Simulationsdaten aus der Datenbank aus und lädt diese herunter. In der zweiten Phase werden die Simulationsdaten in den aktuellen Matlab-Workspace eingetragen. Um Simulationsdaten aus der Datenbank laden zu können, muss sich der Benutzer am Server angemeldet haben. Danach erhält er eine Übersicht über alle Speicherorte, auf die er Zugriff hat. Nachdem er einen Speicherort ausgewählt hat, wird ihm angezeigt, welche Simulationsdaten an dem ausgewählten Speicherort abgelegt sind.

5.3 Informationsmanagement

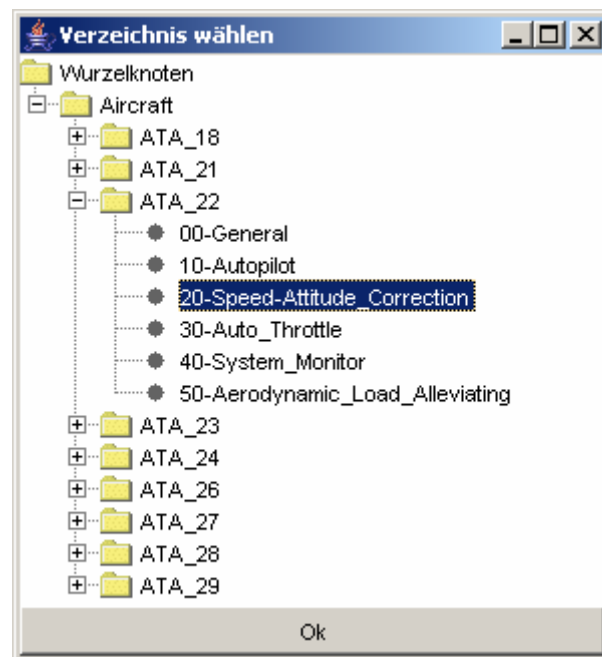


Abbildung 5-37 ATA-Kapitelstruktur zur Ablage von Daten

Zusätzlich zur Anbindung von Matlab wurde eine generische Anwendung implementiert, die in der Lage ist, Simulationsdaten aus einem proprietären Format in das XML-Format zu konvertieren und an den Server zu senden. In der Anwendung können beliebig viele Datei-Parser integriert werden, sodass es möglich ist, beliebige Simulationsprogramme an das verteilte System anzubinden. Exemplarisch wurde ein Dateiparser integriert, der Fluent-Boundary-Profiles lesen und umwandeln kann. In dieses Format werden im CFD-Simulationsprogramm Fluent die Simulationsergebnisse gespeichert; hierbei handelt es sich um einen Zeichenkette (engl.: string) [50]. Für die Konvertierung wurde ein Algorithmus entworfen, der den String iterativ von vorn nach hinten durchläuft, syntaktische Einheiten (engl.: token) zusammenfasst und in einer Datenstruktur ablegt. Die Aufteilung in syntaktische Einheiten wurde durch den Einsatz eines Zustandsautomaten realisiert. Für diese Zerlegung ist es notwendig, dass alle Tokens durch dasselbe Terminalsymbol getrennt werden. In diesem Fall wurde das Leerzeichen ausgewählt, weil für die Zerlegung eines Strings nach Leerzeichen bereits eine Java-Klasse existiert. Die Abbildung 5-38 zeigt das Zustandsdiagramm des Datei-Parsers für Fluent-Dateien. Für die Darstellung des Zustandsautomaten wurde die Modellierungssprache SysML verwendet.

Gesamtkonzeptevaluierung

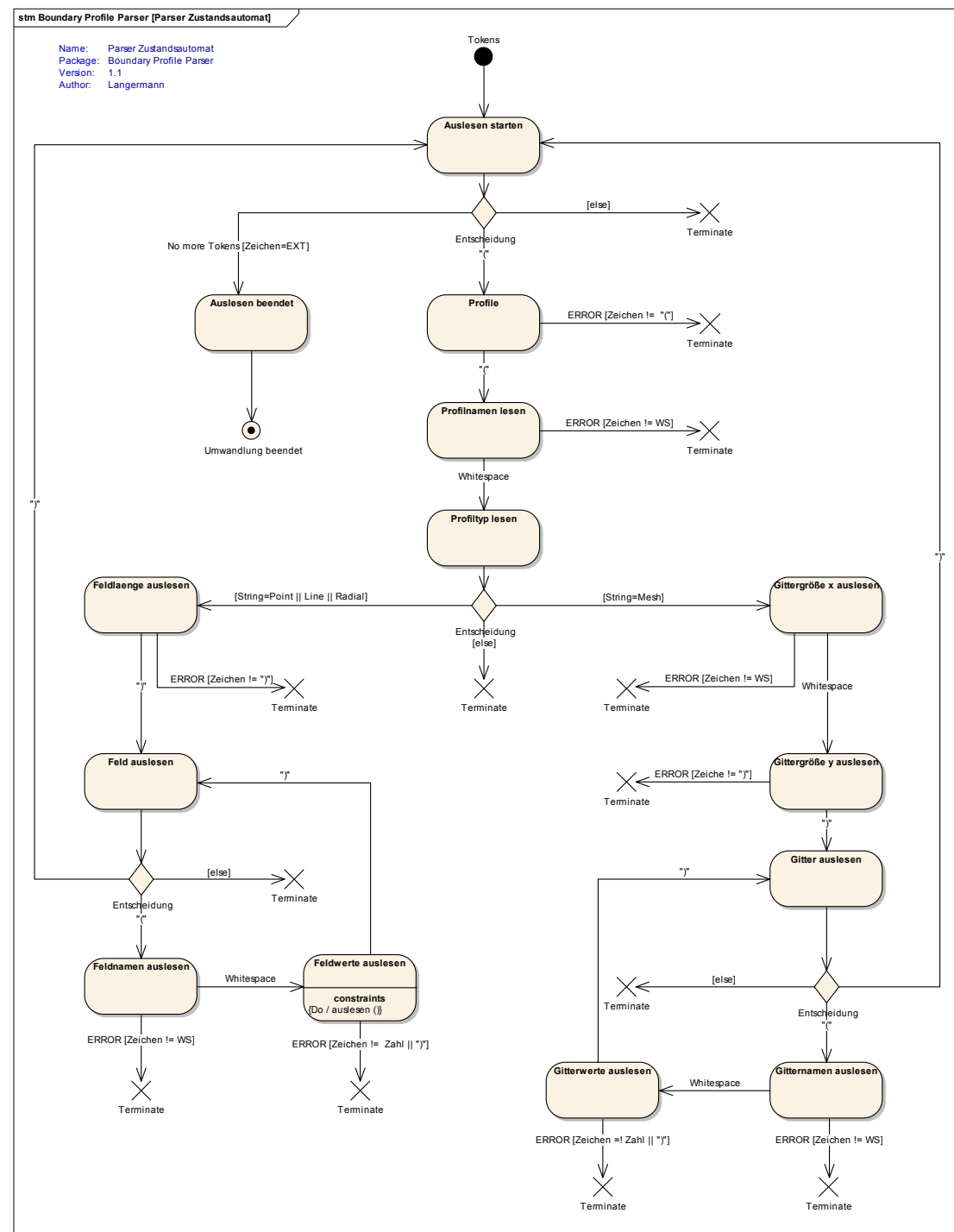


Abbildung 5-38 Zustandsautomat für die Fluent-Boundary-Profile-Parser in SysML

Zunächst wird vom Zustandsautomaten der Name des Profils ausgelesen, anschließend wird der Profiltyp ausgelesen. Das Fluent-Boundary-Profile besteht aus beliebig vielen Datenreihen (engl.: array; Profiltyp point, line oder radial) und Matrizen (Profiltyp mesh). In Abbildung 5-39 ist der Datenfluss im Parser dargestellt. Die Datei wird als Textdatei eingelesen und in eine Liste von Tokens umgewandelt. Der Zustandsautomat arbeitet jeden Token einzeln ab und speichert die Ergebnisse in einer Datenstruktur, die auf Fluent-Boundary-Profiles zugeschnitten ist. Anschließend erfolgt die Konvertierung in einen DOM-Baum und damit die Umwandlung in das XML-Format.

5.3 Informationsmanagement

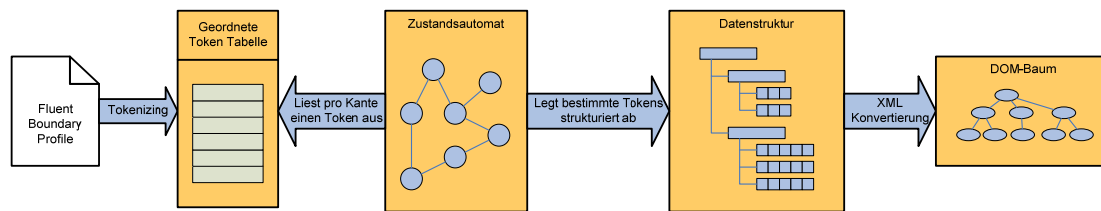


Abbildung 5-39 Datei-Parser für Fluent-Boundary-Profiles

Zusammenfassend lässt sich sagen, dass durch die standardisierte Bereitstellung von Simulationsdaten in einem verteilten System der Austausch von Simulationsdaten zwischen Simulationsprogrammen verbessert wird. Zudem führt die Verwendung eines standardisierten Austauschformats dazu, dass nicht jedes Programm die Konvertierung in jedes proprietäre Datenformat beherrschen muss und somit der Implementierungsaufwand für lose Simulationsskopplungen reduziert wird. Des Weiteren wurde gezeigt, dass nutzerspezifische Zugriffsrechte eine Informationsüberflutung des Benutzers verhindern und zudem die strukturierte Ablage in einem DBMS unterstützen.

6 Zusammenfassung und Ausblick

Der Einsatz von Simulationstechniken hat sich in nahezu allen technischen Bereichen etabliert. Die wichtigsten Argumente für den Einsatz der Simulation sind die Steigerung der Entscheidungsqualität und -sicherheit. Aus wirtschaftlicher Sicht ist daher vor allem bei komplexen Systemen die Simulation in allen Entwicklungsphasen anzuraten. In der Praxis heutiger Entwicklungsprozesse bietet sich allerdings ein anderes Bild. Von einem durchgängigen Simulationseinsatz zur Unterstützung des Entwicklungsprozess kann in der Flugzeugindustrie nicht die Rede sein. Aufgrund von Ressourcenknappheit wird in der Phase der frühen Produktentwicklung häufig auf den Einsatz von Simulationstechniken verzichtet. Gerade diese Phase ist für eine erfolgreiche Produktentwicklung von besonderer Bedeutung, da in dieser Phase bereits bis zu 70 % der Entwicklungskosten festgelegt werden. Es ist daher von entscheidender Bedeutung, den Entwicklungsaufwand in der frühen Phase zu intensivieren, um eine optimale Konzeptauswahl zu gewährleisten. Wird sich zu früh auf ein Konzept festgelegt, ohne weitere Alternativen in Betracht zu ziehen, können in der folgenden Detaillierung grundsätzliche Mängel des Konzepts kaum ausgeglichen werden. Kosten- und zeitintensive Änderungsschleifen sind die Folge.

Im derzeitigen Entwicklungsprozess werden für jedes zu simulierende Problem einzelne in sich geschlossene und nicht wiederverwendbare Simulationsmodelle erstellt. Für den Entwicklungsprozess hat das zur Folge, dass die an der Produktentwicklung beteiligten Fachdisziplinen Insellösungen generieren. Der Blick für das Gesamtsystem geht verloren, und Systeminteraktionen werden nicht berücksichtigt. Eine optimale Gesamtsystemlösung wird sich allerdings niemals aus der Summe der optimalen Teilsystemlösungen ergeben.

Die Ergebnisse von Simulationsstudien sind im Wesentlichen vom verwendeten Datenmaterial, welches für die Erstellung der Modelle und die Eingabe der Simulationsparameter eingesetzt wird, abhängig. Im derzeitigen Entwicklungsprozess stehen benötigte Daten oftmals nicht zur Verfügung oder müssen mit erheblichem Aufwand zunächst beschafft werden. Dies hat zur Folge, dass im Vergleich zur wertschöpfenden Tätigkeit der Aufwand für die Datenbeschaffung sehr hoch ist. Für den effizienten Einsatz der Simulationstechnik ist aus diesem Grund ein geeignetes Informationsmanagementsystem erforderlich.

Die derzeitigen Vorgehensmodelle in der Entwicklung weisen einen gemeinsamen Nachteil auf: Es bestehen zwar Lösungsansätze zur frühen virtuellen Verifizierung der Entwicklungsergebnisse, allerdings fehlt eine konkrete Umsetzung. Aus diesem Grund wird in dieser Arbeit ein Konzept für die durchgängige Simulationsunterstützung im Entwicklungsprozess von Flugzeugsystemen bereitgestellt, das regelt wann und wie die Simulationstechnik im Prozess zu verwenden ist. Basierend auf den Defiziten des Istprozesses wurden Anforderungen an ein Konzept definiert. Aus den Anforderungen wurde ein Gesamtkonzept, bestehend aus den drei Einzelkonzepten für die Informationsbereitstellung, den Simulationseinsatz und die Modellierung, entwickelt. Das Simulationskonzept schreibt den kontinuierlichen Einsatz der Simulationstechnik von der Konzeptphase bis zum Ende der Testphase vor. Ein Modellierungskonzept mit verschiedenen Detaillierungsebenen ermöglicht die Anpassung der Modelle an den jeweiligen Entwicklungsstand und unterstützt damit die Anforderung eines durchgängigen Simulationseinsatzes. Bedingt durch die Simulationsunterstützung in der frühen Entwicklungsphase wird die den Anforderungen entsprechend beste Konzeptalternative für die Weiterentwicklung identifiziert. Die frühzeitige virtuelle Verifikation im linken Strang des V-Modells führt dazu, dass Entwicklungsfehler frühzeitig entdeckt werden, und verhindert damit kostenintensive Iterationsschleifen. Eine Kombination aus alleinstehenden und systemübergreifenden Simulationen ermöglicht zum einen eine Teilsystemoptimierung mit geringem Ressourceneinsatz, und zum anderen werden in den systemübergreifenden Simulationen Systeminteraktionen analysiert, sodass Inkompatibilitäten frühzeitig erkannt werden. Die systemübergreifenden Simulationen werden nach der Konzeptphase und nach der Detaillierungsphase zur Überprüfung des Gesamtsystems eingesetzt. Zur Durchführung der systemübergreifenden Simulationen wird die Simulationskopplung oder die Kosimulation verwendet. Hierdurch wird gewährleistet, dass immer die am besten geeigneten Simulationsprogramme eingesetzt werden können. Der Einsatz der standardisierten Modellierungssprache SysML erhöht zusätzlich die Transparenz und führt zu einer Verbesserung der Kommunikation im Entwicklungsprozess. Gleichzeitig wird durch die standardisierte Modellierung das Verständnis für das Gesamtsystem bei allen an der Entwicklung Beteiligten erhöht.

6 Zusammenfassung und Ausblick

Eine geeignete Informationsbereitstellung erhöht die Produktivität in der Entwicklung und stellt eine Grundvoraussetzung für den kontinuierlichen Einsatz von Simulationstechniken im Entwicklungsprozess dar. Das Gesamtkonzept sieht hierfür eine zentrale Informationsbereitstellung vor. Eine zentrale Informationsbasis ermöglicht die dezentrale Verteilung der Entwicklungsdaten. Dieses Konzept erlaubt es, bestehende Infrastrukturen weiterzuverwenden sowie weitere Informationsquellen zu integrieren, und ermöglicht damit, dass immer die am besten geeignete IT-Lösung für die Informationsbereitstellung eingesetzt wird. Die zentrale Informationsbereitstellung hat den Vorteil, dass sich der Nutzer nur einmal authentifizieren muss und über eine einheitliche grafische Benutzerschnittstelle auf alle verfügbaren Entwicklungsdaten zugreifen kann. Durch die zentrale Vergabe individueller Zugriffsrechte wird eine Informationsüberflutung des Nutzers verhindert, und zudem wird die intuitive Navigation in der IB unterstützt. Über automatische Benachrichtigungen werden den Mitarbeitern die aktuellen Informationen bereitgestellt. Die standardisierte Ablage von Modellen, Dokumenten und Daten erhöht die Wiederverwendung und Verbesserung der Kommunikation. Eine Intensivierung des systemübergreifenden Informationsaustausches führt damit zu einer Reduzierung der Iterationsschleifen in der Entwicklung. Zudem wird der Aufwand für die Beschaffung von Informationen reduziert und somit die wertschöpfende Tätigkeit aller Beteiligten gesteigert.

Die qualitativen Vorteile des Gesamtkonzepts wurden anhand einer Brennstoffzellensystementwicklung quantifiziert. Durch den Einsatz der Simulation in der Konzeptfindungsphase war es möglich, verschiedene Brennstoffzellensystemkonzepte zu vergleichen. Es stellte sich heraus, dass zum gegenwärtigen Stand der Technik eine Substitution konventioneller Flugzeugsysteme durch Brennstoffzellensysteme nicht sinnvoll ist. Unter ökonomischen und ökologischen Gesichtspunkten kann sich der Einsatz von Brennstoffzellensystemen mittelfristig durchaus rechnen. Allerdings ist hierfür eine Verbesserung der Brennstoffzellen- und Reformierungstechnologien erforderlich. Des Weiteren wurde für die Umsetzung der Informationsbasis ein Master-Server-System vorgestellt. Ein standardisiertes XML-Format wurde für die lose Simulationskopplung und eine verteilte Simulation auf Basis der HLA für die feste Simulationskopplung dargestellt. Für die Umsetzung des Gesamtkonzepts wurden frei erhältliche Softwareprogramme eingesetzt (z. B. MySQL, RTI 1.3-NG), für die Realisierung des vorgestellten Konzepts in der Industrie muss über den Einsatz kommerzieller Programme nachgedacht werden. Dies ist aus Gründen der Wartung sowie der Softwareunterstützung von Bedeutung.

Der Einsatz von SysML wird von der Flugzeugindustrie forciert. Für die Aufnahme funktionaler Anforderungen, zur Modellierung und für die Simulation von Zustandsautomaten wird bei Airbus bereits die SysML-basierte Software Rhapsody des Unternehmens Telelogic eingesetzt. Zum gegenwärtigen Zeitpunkt existieren noch keine kommerziellen Lösungen für den Einsatz der SysML in kontinuierlichen Simulationsprogrammen. Verschiedene Simulationssoftwarehersteller arbeiten bereits an dieser Implementierung, sodass die Modellierung kontinuierlicher Simulationsstudien mit der SysML nur eine Frage der Zeit ist. Des Weiteren wird die kontinuierliche Leistungssteigerung der Arbeitsplatzrechner es langfristig erlauben, auch gestaltorientierte Echtzeitsimulationen durchzuführen.

Diese Arbeit liefert einen Beitrag zur Realisierung einer virtuellen Flugzeugentwicklung. Das Ziel ist eine frühzeitige virtuelle Analyse des Verhaltens des Gesamtsystems Flugzeug. Dies hat eine Reduzierung der realen Hardwaretests zur Folge. Zudem erhofft man sich, dass durch eine virtuelle Überprüfung der Aufbau eines sogenannten Iron Birds (Flugzeugprüfstand: stationärer Aufbau zum Testen der Systeme eines neuen Baumusters) obsolet wird. Aus der frühzeitigen virtuellen Flugzeugüberprüfung resultiert eine Reduzierung der Entwicklungszeit und -kosten.

7 Anhang

7.1 Mathematische Grundlagen der Simulationstechnik

In diesem Kapitel werden die mathematischen Grundlagen zum Lösen von linearen und nichtlinearen Gleichungssystemen sowie von Differenzialgleichungen vorgestellt.

Lösen von Gleichungssystemen mit mehreren Unbekannten

Für das Lösen von linearen Gleichungssystemen eignet sich das gaußsche Eliminationsverfahren [111]. Es ist ein Algorithmus aus den mathematischen Teilgebieten der linearen Algebra und der Numerik. Im ersten Schritt wird das Gleichungssystem durch Äquivalenzumformungen in die Dreiecksmatrix gebracht. Im zweiten Schritt werden, ausgehend von der letzten Zeile, in der sich nur noch eine Variable befindet, die Variablen ausgerechnet und in die darüberliegende Zeile eingesetzt (vgl. [109]). Weitere Verfahren werden im Bärwolf [7] beschrieben.

Lösung von Differenzialgleichungen

Eine Differenzialgleichung (DGL) ist eine Gleichung, in der eine gesuchte Funktion und deren Ableitungen nach einer oder mehreren Variablen auftreten. In allgemeiner Form lässt sich eine Differenzialgleichung in folgender Form darstellen [75].

$$y^n(x) = F(x, y, y', \dots, y^{(n-1)}) \quad (7-1)$$

Zum Lösen von Differenzialgleichungen bietet die Mathematik mehrere Verfahren an. Es folgen drei dieser Verfahren [21]:

- Laplace-Transformation
- Analytische Verfahren
- Zustandsgleichung, Zustandsform

Bei der Laplace-Transformation handelt es sich um eine Integraltransformation, mit der eine DGL mit konstanten Koeffizienten in eine algebraische Gleichung umgewandelt wird, wodurch sich das Lösen vereinfacht [187]. Die analytischen Verfahren führen zu mathematisch geschlossenen Lösungen. Für weitere Informationen zu diesen Verfahren sei an dieser Stelle auf folgende Standardwerke verwiesen [75], [76], [60], [61]. Durch das Umformen der DGL in die Zustandsform kann eine Differenzialgleichung n-ter Ordnung in ein Differenzialgleichungssystem mit n Differenzialgleichungen 1. Ordnung überführt werden [15], [115]. Ergänzend zu diesen Verfahren stellt die numerische Mathematik weitere Lösungsverfahren zur Verfügung.

Numerische Methoden zum Lösen von DGL

In diesem Abschnitt werden einige wichtige numerische Lösungsverfahren gewöhnlicher Differenzialgleichungen beschrieben. Für die Implementierung zeitkontinuierlicher Systeme auf Digitalrechnern bietet sich zur Umsetzung von Problemen 1. Ordnung der Form

$$y'(x) = f(x, y(x)) \quad (7-2)$$

die Euler-Methode geradezu an. Diese basiert auf der Taylor-Reihenentwicklung [20].

$$f(x_0, h) = f(x_0) + \frac{f'(x_0)}{1!} \cdot h + \frac{f''(x_0)}{2!} \cdot h^2 + \dots + \frac{f^{(n)}(x_0)}{n!} + R_n(h) \quad (7-3)$$

Wird die Taylor-Reihe nach dem zweiten Term abgebrochen, erhält man die Integrationsmethode nach Euler.

$$y_{k+1} = y_k + h \cdot f(x_k, y_k) \quad (7-4)$$

7.1 Mathematische Grundlagen der Simulationstechnik

Für die Berechnung des folgenden Näherungswertes y_{k+1} wird die Steigung in approximierten Punkten (x_k, y_k) benutzt [7]. Bei diesem Verfahren wird nur der vorangegangene Wert für die Berechnung des nächsten Wertes benötigt, es handelt sich somit um ein Einschrittverfahren [187]. Der Rechenaufwand für dieses Verfahren ist niedrig, allerdings ist der numerische Fehler von der Ordnung $(h)^2$.

Eine wesentlich genauere numerische Lösung der Gleichung (7-2), stellt das Runge-Kutta-Verfahren 4. Ordnung zur Verfügung. Dieses Verfahren stimmt mit der Taylor-Reihenentwicklung bis zu den Gliedern 4. Ordnung überein, damit ist der numerische Fehler von der Ordnung $(h)^5$. Die höhere Genauigkeit des Runge-Kutta-Verfahrens rechtfertigt den höheren Rechenaufwand. Es ist oftmals möglich, die Schrittweite h zu vergrößern und dadurch Rechenzeit zu sparen (variable Schrittweite) [187].

Ergänzend zu den Einschrittverfahren existieren Mehrschrittverfahren, die den nächsten Wert mithilfe von mehreren vorangegangenen Werten berechnen. In Einschrittverfahren wird Information über den Zustand in einem Zeitpunkt verwendet, um den Zustand in einem späteren Zeitpunkt zu berechnen. In Mehrschrittverfahren wird dazu auf Information aus mehreren früheren Zeitpunkten zugegriffen [85].

Des Weiteren unterscheidet man implizite und explizite numerische Integrationsverfahren. In ist eine Differenzialgleichung erster Ordnung in expliziter und impliziter Form dargestellt.

Explizite Form	Implizite Form
$y'(x) = f(y(x), x)$	$0 = f(y'(x), y(x), x)$

Tabelle 7-1 Implizite und explizite Form

Es wird deutlich, dass die explizite Form direkt gelöst werden kann, während die implizite Form Iterationsschritte erfordert.

In Tabelle 7-2 sind die bekanntesten numerischen Integrationsverfahren angeführt. Für weitere Einzelheiten zu den aufgelisteten Verfahren sei an dieser Stelle auf folgende Quellen verwiesen [7], [75].

Verfahren	Typ
Euler (explizit)	Einschrittverfahren
Euler (implizit)	Einschrittverfahren
Trapez-Verfahren (implizit)	Einschrittverfahren
Runge-Kutta (explizit)	Einschrittverfahren
Runge-Kutta (explizit)	Mehrschrittverfahren
Runge-Kutta (implizit)	Einschrittverfahren
Adam-Moulton (implizit)	Mehrschrittverfahren
Adam-Bashforth (explizit)	Mehrschrittverfahren
BDF (Rückwärtsdifferenzen-Verfahren)	Mehrschrittverfahren

Tabelle 7-2 Numerische Integrationsverfahren

Die Verfahren sind für Differenzialgleichungen erster Ordnung anwendbar. Da sich eine Differenzialgleichung höherer Ordnung auf ein Differenzialgleichungssystem erster Ordnung zurückführen lässt, sind die vorgestellten Integrationsverfahren universell für die Lösung von linearen und nichtlinearen Differenzialgleichungen einsetzbar [187].

Steifigkeit der DGL

Für die Auswahl eines geeigneten numerischen Integrationsverfahrens aus Tabelle 7-2 ist es wichtig zu wissen ob es sich um ein steifes System handelt. Im Allgemeinen gilt für steife Systeme, dass die Jacobi-Matrix negative Eigenwerte mit großen Gradienten aufweist [85]. Nach Angermann lautet die Definition für steife Systeme folgendermaßen: „Eine Differenzialgleichung wird als steif bezeichnet, wenn das charakteristische Polynom gleichzeitig sehr kleine und sehr große Nullstellen besitzt“ [4]. Vereinfacht gilt für ein steifes Problem, dass im System sehr stark unterschiedliche Zeitkonstanten enthalten sind; gleichzeitig laufen im System also einige Vorgänge sehr schnell und andere sehr langsam ab [163].

Anhang

Klassen von Differenzialgleichungen

Neben den bereits beschriebenen gewöhnlichen Differenzialgleichungen (Abk. ODE; engl.: Ordinary Differential Equation), die nur Ableitungen nach einer Veränderlichen enthalten, existieren weitere Differenzialgleichungen. In differenzialalgebraischen Gleichungen (Abk. DAE; engl.: Differential Algebraic Equation) sind algebraische Gleichungen mit Differenzialgleichungen gekoppelt. Die partiellen Differenzialgleichungen stellen die höchsten Anforderungen an die Lösungsverfahren. Sie müssen z. B. mit den Verfahren der finiten Elemente gelöst werden [15].

Linearisierung (Newton-Verfahren)

In einer linearen Gleichung kommt die Unbekannte ausschließlich in der ersten Potenz vor. Bei Gleichungen höheren Grades (z. B. quadratisch, kubisch usw.) spricht man von nichtlinearen Gleichungen. Ein Problem nichtlinearer Gleichungen sind mehrdeutige Lösungen [vgl. Gleichung (7-5)].

$$\begin{aligned} x^2 + y^2 &= 1 \\ y_{1,2} &= \pm \sqrt{1 - x^2} \end{aligned} \quad (7-5)$$

Mit einem numerischen Algorithmus erhält man je nach Initialisierungswert eine dieser Lösungen [26]. Im konkreten Fall einer Simulation kann das bedeuten, dass mit einem falschen Wert weitergerechnet wird.

Für Lösungen algebraischer Gleichungen bis zur 4. Ordnung existieren analytische Methoden (z. B. pq-Formel für quadratische Probleme), für Gleichungen höherer Ordnung werden Iterationsverfahren verwendet. Dabei handelt es sich meist um Verfahren zur Bestimmung von Nullstellen.

Eines der am häufigsten verwendeten Iterationsverfahren für die Ermittlung reeller Nullstellen von algebraischen Gleichungen ist das Newton-Verfahren. Es basiert auf der Taylor-Reihenentwicklung [187].

$$g(x) = f(x_0) + f'(x_0) \cdot (x - x_0) = 0 \Rightarrow x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (7-6)$$

Unter bestimmten Voraussetzungen konvergiert diese Folge gegen eine Nullstelle. Da die Steigung $f'(x_0)$ für die Schrittweite $(x - x_0)$ als konstant angenommen wird, spricht man auch von einer Linearisierung. Dieses Verfahren eignet sich auch für nichtlineare Gleichungssysteme. Man kann z. B. den Schnittpunkt von zwei Gleichungen, und damit die Lösung einer Variablen für die Gleichungen, mit dem Newton-Verfahren bestimmen. Dafür subtrahiert man eine Gleichung $f_1(x)$ von der anderen $f_2(x)$ und erhält eine neue Gleichung $f_3(x)$. Für diese wird mithilfe des Newton-Verfahrens eine Nullstelle ermittelt [Abbildung auf Nullstellenproblem; vgl. Gleichung (7-7)] [32].

$$\begin{aligned} f_1(x) &= 2x + 2; \quad f_2(x) = e^{-x} \\ f_3(x) &= e^{-x} - 2x + 2 \\ f_3(x) &= 0 \rightarrow x \approx -0,3 \end{aligned} \quad (7-7)$$

Jacobi-Matrix

Die Linearisierung von nichtlinearen Gleichungssystemen kann mithilfe der Jacobi-Matrix durchgeführt werden. Dabei wird, basierend auf der Taylor-Reihenentwicklung, das nichtlineare Gleichungssystem um einen Bezugspunkt linearisiert. Auf das linearisierte Gleichungssystem lassen sich anschließend alle Methoden der linearen Analyse anwenden (z. B. gaußsche Eliminationsverfahren) [15].

Die Jacobi-Matrix ist eine mehrdimensionale Erweiterung der eindimensionalen Ableitung [111]. Sie stellt eine lineare Approximation einer differenzierbaren Funktion in einem gegebenen Punkt dar und bildet somit die Matrixdarstellung der partiellen Ableitungen (Gradienten) dieser Funktion.

7.1 Mathematische Grundlagen der Simulationstechnik

$$J = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (7-8)$$

Damit ist die Jacobi-Matrix eine Matrix linearisierter Funktionssegmente. Diese Segmente werden für die Ermittlung der Nullstellen benötigt. Die folgende Gleichung (7-9) ist eine multidimensionale Darstellung des bereits erwähnten Newton-Verfahrens (siehe Gleichung (7-6)) [32].

$$\Delta \vec{x} = -J^{-1} \cdot f(x) \quad (7-9)$$

Algebraische Schleifen

Algebraische Schleifen entstehen durch das direkte Rückführen von Ausgangsgrößen auf den Eingang. Das bedeutet, dass die zu berechnende Größe direkt von der Eingangsgröße abhängt. Die Eingangsgröße muss also bekannt sein, um die Ausgangsgröße berechnen zu können. In Abbildung 7-1 ist ein Signalflussbild mit direkter Rückführung für eine Subtraktionsstelle dargestellt. Es wird deutlich, dass die Ausgangsgröße Z aus sich selbst berechnet wird.

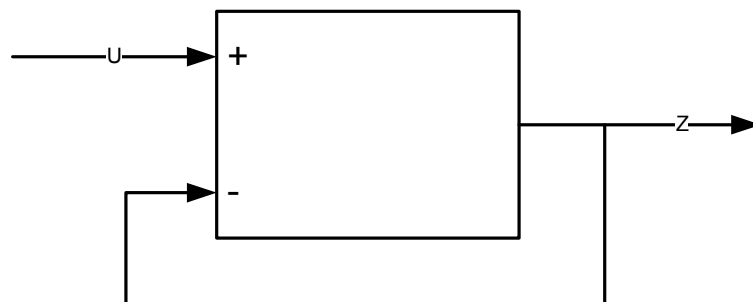


Abbildung 7-1 Beispiel einer algebraischen Gleichung

Aufgrund der sequenziellen Abarbeitung durch den Rechner können algebraische Schleifen Konvergenzprobleme verursachen. Aus diesem Grund sollten sie bei der Modellierung vermieden werden.

Gleichgewichtspunkte

Gleichgewichtspunkte sind Ruhepunkte eines Systems, in ihnen verschwinden die Ableitungen der Zustandsgrößen nach der Zeit ($dx/dt = 0$ eingeschwungener Zustand) [15], [4]. Damit vereinfacht sich ein Differenzialgleichungssystem zu einem algebraischen Gleichungssystem. Ein Gleichgewichtspunkt zeichnet sich dadurch aus, dass das System bei konstantem Eingang für alle zukünftigen Zeiten in einem Zustand verharrt, sobald es diesen Zustand erreicht hat. Dies gilt für alle linearen und nichtlinearen Systeme. Für kontinuierliche lineare Systeme ist der Zeitpunkt $t = 0$ immer der einzige Gleichgewichtspunkt [15].

Symbolische Mathematik

Für das Lösen von Gleichungen können die numerische Iteration [z. B. Newton-Verfahren, vgl. Gleichung (7-6)] sowie die symbolische Gleichungslösung genannt werden. Wie bereits im Abschnitt Linearisierung beschrieben, sind iterative numerische Verfahren prinzipiell sehr einfach, haben jedoch den Nachteil, dass nur eine von eventuell mehreren mathematisch korrekten Lösungen gefunden werden kann [vgl. Gleichung (7-5)].

Anhang

Die symbolische Gleichungslösung erfolgt nach einem dem Gauß-Algorithmus ähnlichen Verfahren, das auch auf nichtlineare Gleichungen angewendet werden kann. Mit diesem Verfahren werden alle mathematisch korrekten Lösungen gefunden [3]. Für weitere Informationen sei an dieser Stelle auf ein Standardwerk für die symbolische Behandlung von Differenzialgleichungen, den Kamke, verwiesen [75], [76].

7.2 Gestaltorientierte Simulation

In diesem Kapitel werden die verschiedenen Einsatzmöglichkeiten und Typen von gestaltorientierten Simulatoren vorgestellt. Da die „Gestalt“ und damit z. B. die Anordnung von Systemkomponenten Auswirkungen auf das Systemverhalten hat, werden diese Simulatoren ebenfalls in der Systementwicklung eingesetzt. Für diese Art der Simulation sind genaue Kenntnisse von Randbedingungen, Material und Geometrie erforderlich. Gestaltorientierte Simulatoren werden in der Berechnung von Festkörper, Strukturen, Wärmeproblemen, magnetischen Potenzialen sowie in der Lichtsimulation, Strömungsmechanik, Crash-Simulation und in vielen weiteren Bereichen eingesetzt. Als Berechnungsgrundlage für diese Simulatoren dient die Diskretisierung von Oberflächen oder Volumina. Die Berechnungsgebiete werden dafür in endliche (lat.: finis) Bereiche aufgeteilt. Die wichtigsten Methoden, die nach der Diskretisierung des Integrationsgebietes direkt eine näherungsweise Lösung liefern, sind:

- Finite-Elemente-Methode (FEM)
- Finite-Volumen-Methode (FVM)
- Finite-Differenzen-Methode (FDM)

Für die Finite-Differenzen-Methode wird der Integrationsbereich in orthogonale Gitter diskretisiert. Vor der Anwendung für komplexe Konfigurationen ist eine aufwendige Transformation notwendig. Diese entfällt bei der FVM und der FEM. Die Finite-Elemente-Methode besitzt die höchste Flexibilität, da sie auf flexiblen unstrukturierten Netzen aufbaut [121].

In Abbildung 7-2 sind die im Folgenden beschriebenen numerischen Lösungsmethoden bezüglich ihrer Genauigkeit und Flexibilität dargestellt.

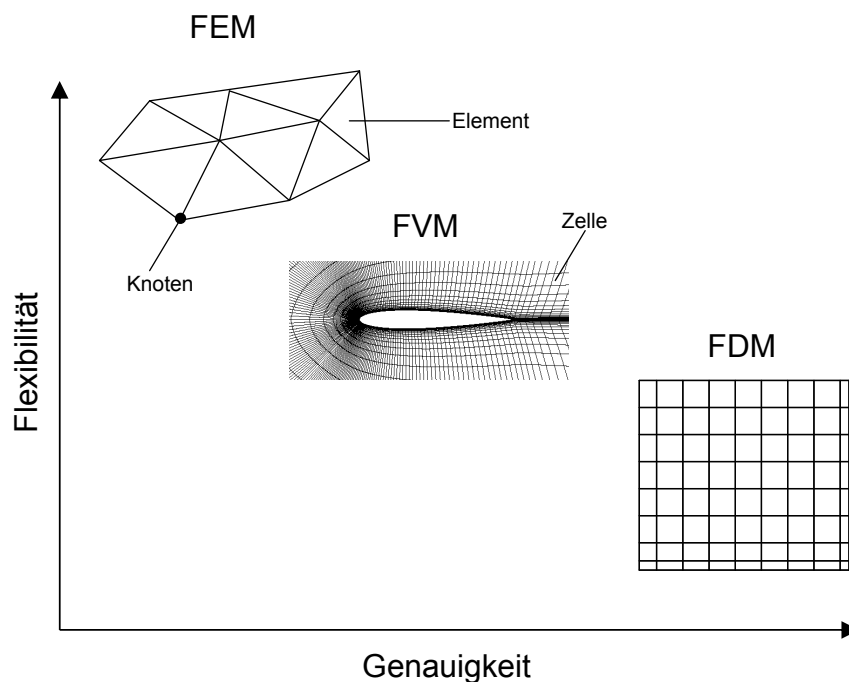


Abbildung 7-2 Genauigkeit und Flexibilität numerischer Lösungsmethoden [121]

Finite-Elemente-Methode

Die Methode der finiten Elemente wurde ursprünglich in der Festkörpermechanik zur Berechnung von Strukturproblemen entwickelt [66]. Im ersten Schritt wird das Lösungsgebiet in sich nicht überlappende geometrische Elemente gleichen Typs aufgeteilt. Die Eckpunkte der Elemente werden als Knoten bezeichnet. Alle Knoten und Elemente zusammen bilden ein Netz, welches das Lösungsgebiet diskretisiert [121].

Für diese Elemente werden Ansatzfunktionen definiert, aus denen sich über Differenzialgleichungen und Randbedingungen ein Gleichungssystem ergibt. Letztendlich ergibt sich ein algebraisches Gleichungssystem, mit dessen Hilfe die Näherungslösungen der gesuchten Funktion in den Knotenpunkten des Gitters ermittelt werden [66]. Für weitere

Anhang

Einzelheiten zu diesem Verfahren sei an dieser Stelle auf einschlägige Literatur verwiesen [8].

Finite-Differenzen-Methode

In Analogie zur FEM wird bei der Finite-Differenzen-Methode der Integrationsraum zunächst diskretisiert. In einem Beispiel wird hierfür die kontinuierliche Zeit t in äquidistante Zeitintervalle Δt unterteilt. An den Intervallgrenzen ist der jeweilige Funktionswert zu bestimmen (vgl. Abbildung 7-3).

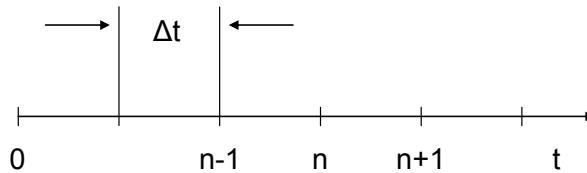


Abbildung 7-3 Zeitliche Diskretisierung [121]

Anschließend werden die Differenzialgleichungen in diskreten Gitterpunkten in Differenzengleichungen umgewandelt. Wie in Abbildung 7-2 dargestellt, setzt dieses Vorgehen ein orthogonales Netz voraus. Ausgehend von einem Anfangswertproblem zum Zeitpunkt $t = 0$ kann mithilfe der Taylor-Reihenentwicklung [vgl. Gleichung (7-3)] und mit Kenntnis der Ableitung an der Stelle $t = t_n$ der zukünftige Funktionswert berechnet werden [121].

$$u^{n+1} = u^n + \Delta t \cdot \frac{\partial u^n}{\partial t} \quad (7-10)$$

Bricht man die Taylor-Reihe nach dem zweiten Glied ab [vgl. Gleichung (7-10)] so ergibt sich das explizite Euler-Verfahren. Die Ordnung des Verfahrens ergibt sich aus dem Term, an dem die Taylor-Reihe abgebrochen wurde. Verfahren höherer Ordnung sind genauer, erfordern aber einen erhöhten Rechenaufwand pro Iteration, dafür jedoch weniger Iterationen. Es existieren zu diesem Thema zahlreiche weitere implizite und explizite Verfahren; für weitere Informationen sei an dieser Stelle auf Quelle [7] verwiesen.

Finite-Volumen-Methode

Ähnlich wie bei den beiden bereits beschriebenen Methoden wird das Lösungsgebiet mithilfe von numerischen Netzen diskretisiert. Bei der FVM werden allerdings die Erhaltungsgleichungen über das jeweilige Volumenelement in integraler Form erfüllt. Ein Erhaltungssatz ist durch die Gleichung

$$\frac{\partial}{\partial t} u + \nabla \cdot f(u) = 0 \quad (7-11)$$

im Lösungsbereich gegeben.

Im Unterschied zu dem Ausdruck „Element“ wird in der Finite-Volumen-Methode der Ausdruck „Zelle“ benutzt. In Abbildung 7-4 sind verschiedene Zelltypen dargestellt. Ein weit verbreiteter Einsatzbereich der Finite-Volumen-Methode liegt in der numerischen Strömungsmechanik, in der sie als Standardverfahren zur Lösung kompressibler Strömungsprobleme, also der Euler- und Navier-Stokes-Gleichungen, benutzt wird. Den sogenannten Computational Fluid Dynamics-Programmen (Abk. CFD) ist der folgende Abschnitt gewidmet. Für Einzelheiten zur Finite-Volumen-Methode siehe [121].

7.2 Gestaltorientierte Simulation

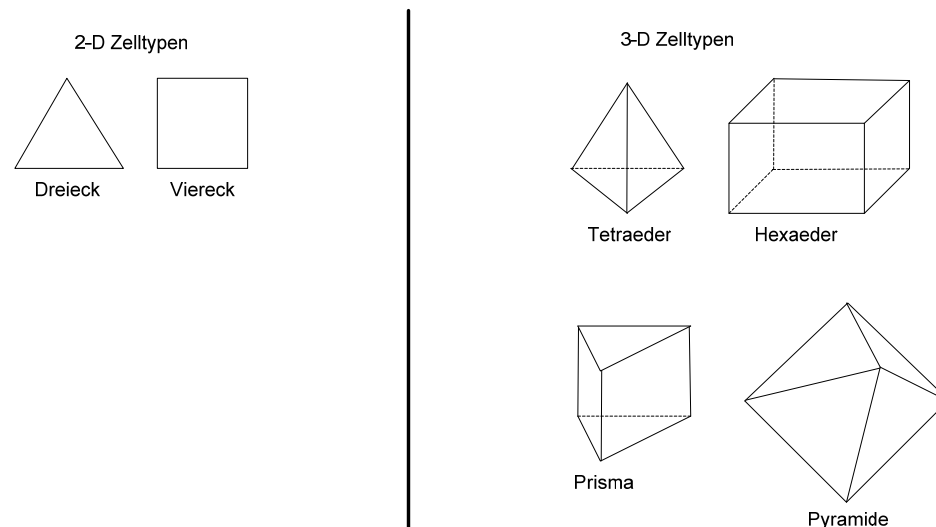


Abbildung 7-4 Zellelemente

Computational Fluid Dynamics (CFD)

Die Beschreibung der numerischen Strömungssimulation als Methode der finiten Volumen ist das Thema in diesem Abschnitt. Sie ist für das Verständnis der Arbeit von Bedeutung. So wird z. B. in Kapitel 5.2.3.1 auf die lose Kopplung von Systemsimulatoren mit CFD-Programmen eingegangen.

Mithilfe der numerischen Strömungsmechanik ist es möglich, strömungstechnische Probleme für newtonsche Fluide, die messtechnisch nur aufwendig erfassbar sind, zu simulieren [170]. Die verwendeten Modellgleichungen gehen von der Kontinuumsannahme aus. Grundlage der mathematischen Modellierung von Transportprozessen sind Bilanzgleichungen. Dies basiert auf der Überlegung, dass die Speicherung einer Bilanzgröße in einem Volumenelement gleich der Summe der Ströme der Bilanzgrößen durch die Begrenzungsflächen und der Volumenquelle sein muss [127].

Wie bereits im vorangegangenen Abschnitt beschrieben, wird das Lösungsgebiet, in diesem Fall das Strömungsfeld, in endliche Volumina (Zellen; vgl. FVM) geteilt, und die Erhaltungsgleichungen für Impuls, Masse und Energie (Navier-Stokes-Gleichungen) werden an jedem Volumenelement unter der Einhaltung der vorgegebenen Randbedingungen iterativ gelöst (siehe Abbildung 7-5). Die Berechnung mittels CFD liefert dabei an jedem Ort des diskretisierten Strömungsfeldes die Größen für Druck, Geschwindigkeit und Temperatur [170].

Eine wesentliche Erhaltungsgröße in der Strömungsmechanik ist die Masse. Die Bilanz des Massenerhaltes (Kontinuitätsgleichung) lässt sich in vektorieller Form wie folgt schreiben [vgl. auch Gleichung (7-11)] [149].

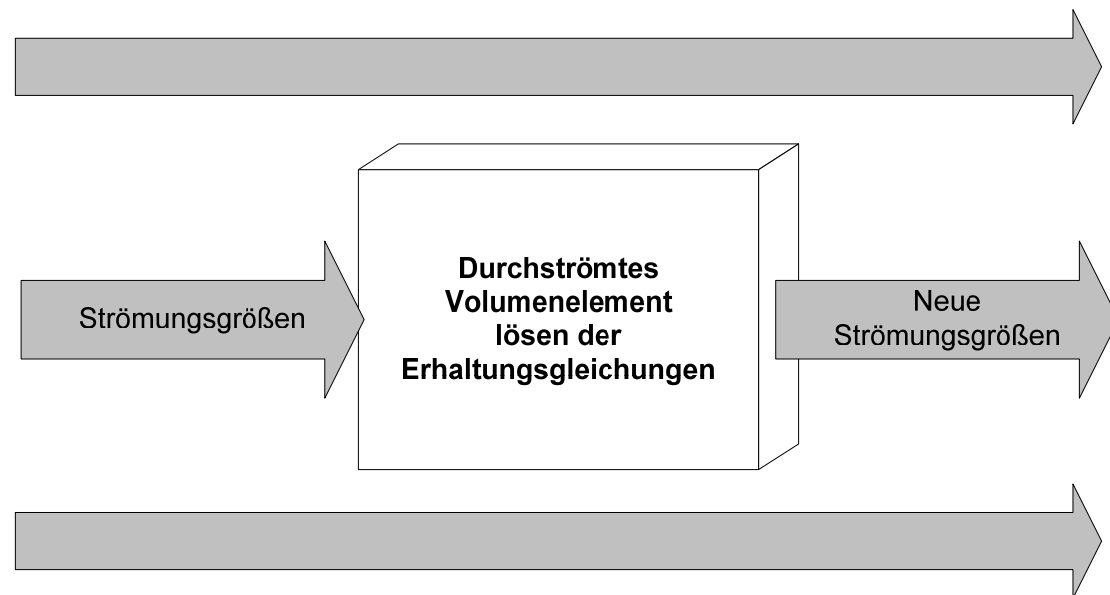
$$\frac{d\rho}{dt} + \nabla(\rho \cdot \vec{v}) = 0 \quad (7-12)$$

wobei ρ die Dichte, \vec{v} den Geschwindigkeitsvektor und ∇ den Nabler-Operator bezeichnet. In Fluid-Systemen treten meist dynamische Vorgänge auf. Aus diesem Grund nimmt die Berechnung des Geschwindigkeitsfeldes eine zentrale Rolle ein. Die Impulsbilanz (Momentumerhaltung) ermöglicht dies. In vektorieller Schreibweise ergibt sich für die Impulsbilanz folgender Zusammenhang [149].

$$\frac{\partial}{\partial t}(\rho \cdot \vec{v}) + \nabla(\rho \vec{v} \otimes \vec{v}) = -\nabla p + \nabla \bar{\tau} + \vec{f} \quad (7-13)$$

wobei p den Druck, $\bar{\tau}$ den Spannungstensor und \vec{f} den Beschleunigungsvektor aufgrund einer äußeren Volumenkraft bezeichnet.

Anhang

**Abbildung 7-5 Durchströmtes finites Volumen**

Um den Spannungstensor zu ermitteln, werden statistische Turbulenzmodelle verwendet, die auf den mit der Reynoldszahl gemittelten Navier-Stokes-Gleichungen (Abk. RANSE; engl.: Reynolds Averaged Navier Stokes Equations) aufsetzen.

Für die Energiebilanz im Volumenelement werden die Terme für ein- bzw. ausströmende Energieströme, die am Volumenelement von außen geleistete Arbeit und aufgebrauchte Energie (z. B. Wärmeleitung) sowie die durch innere Volumenkräfte geleistete innere Arbeit aufgestellt. Für eine detaillierte Herleitung der Energiebilanz sei auf Quelle [127] verwiesen.

$$\frac{\partial}{\partial t}(\rho E) + \nabla(\vec{v}(\rho E + p)) = -\nabla \vec{q} + \frac{\partial Q}{\partial t} + \Phi \quad (7-14)$$

wobei E die innere Energie, \vec{q} den Wärmestromvektor, Q die Wärmeproduktion und Φ die Dissipationsfunktion bezeichnet. Die Dissipationsfunktion stellt dabei die Umwandlung von kinetischer Energie der Strömung in innere Energie dar.

Dieses ergibt, betrachtet man den Impulserhalt in allen drei Raumrichtungen getrennt, fünf Gleichungen zur Berechnung der sechs Unbekannten für den Druck p , die Dichte ρ , die Geschwindigkeiten in den drei Raumrichtungen u , v , w und die Energie E im Volumenelement. Somit müssen an dieser Stelle je nach Art des diskretisierten Strömungsgebiets Hilfgleichungen wie die Zustandsgleichung idealer Gase für ein kompressibles Strömungsgebiet [siehe Gleichung (7-14)] oder im Falle betrachteter Wärmeleitung das Fourier-Gesetzes zum Einsatz kommen [27].

$$p = \rho \cdot R \cdot T \quad (7-15)$$

wobei R die spezifische Gaskonstante des Mediums und T die Temperatur des Gases beschreibt. Für eine detaillierte Herleitung der numerischen Strömungssimulation wird folgende Quelle empfohlen [149].

Dieses Kapitel hat keinen Anspruch auf Vollständigkeit, zeigt aber in welchen verschiedensten technischen Bereichen die gestaltorientierten Simulationsprogramme eingesetzt werden.

7.3 Systemsimulation

In diesem Kapitel werden die drei in Kapitel 2 vorgestellten Systemsimulationsklassen beschrieben. Es werden Gemeinsamkeiten und Unterschiede zwischen ihnen aufgezeigt, und es wird beschrieben, für welchen Einsatz die einzelnen Systemsimulatoren geeignet sind. Anhand einzelner ausgewählter Softwareprogramme sollen die unterschiedlichen Modellierungsansätze und Lösungsverfahren vorgestellt werden.

Im Gegensatz zu den gestalterorientierten Simulatoren wird in der Systemsimulation keine Geometrie und damit keine Diskretisierung des Lösungsbereiches benötigt. Vielmehr wird das Verhalten, also die Funktionsweise der Systemkomponenten, mithilfe von Differenzialgleichungen beschrieben. Die Lösung der Differenzialgleichung erfolgt meist unter Zuhilfenahme der numerischen Mathematik (vgl. Kapitel 7.1).

Die Systemsimulatoren lassen sich nach Brychat in drei Klassen einteilen (vgl. Abbildung 7-6). Sie unterscheiden sich in der Modellierung des zu simulierenden Systems. Alle Programme sind in mathematische Hintergrundprogramme mit leistungsfähigen Gleichungslösern eingebunden und besitzen spezifische Modellbibliotheken, die das Modellieren vereinfachen [21].

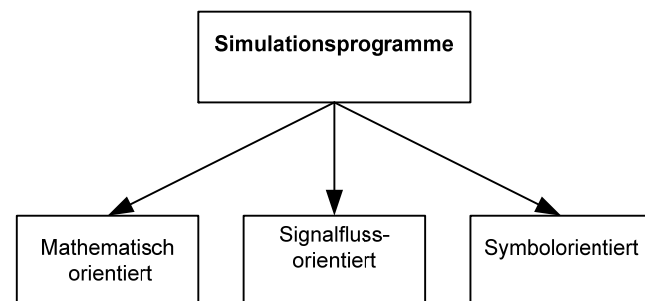


Abbildung 7-6 Klassen von Systemsimulationsprogrammen

In den folgenden drei Kapiteln werden anhand eines Feder-Masse-Systems die unterschiedlichen Modellierungsverfahren der drei Simulationsprogramme grundsätzlich gezeigt. Für das mechanische Ersatzschaltbild ergibt sich die Differenzialgleichung 2. Ordnung.

$$F = x'' \cdot m + x' \cdot d + x \cdot c \quad (7-16)$$

wobei F die Kraft, x'' die Beschleunigung, m die Masse, x' die Geschwindigkeit, d die Dämpfung, x die Auslenkung und c die Federkonstante ist. Zusätzlich werden die Funktionsweise der Simulationsprogramme sowie verschiedene Softwareprogramme vorgestellt.

7.3.1 Mathematisch orientierte Simulationsprogramme

Für mathematisch orientierte Simulationsprogramme wird zunächst das Differenzialgleichungssystem aufgestellt. Für die Lösung stehen, wie bereits erwähnt, die numerische und symbolische Mathematik zur Verfügung (vgl. Kapitel 7.1). Beispielhaft wird im Folgenden die Gleichung (7-16) mithilfe der numerischen Integration gelöst. Die Differenzialgleichung 2. Ordnung wird dafür in zwei Differenzialgleichungen 1. Ordnung überführt.

$$\begin{aligned} x &= x_0, \dot{x} = \dot{x}_0 = x_1, \ddot{x} = \dot{x}_1 \\ \dot{x}_0 &= x_1 \\ \dot{x}_1 &= -x_1 \frac{d}{m} - x_0 \frac{c}{m} + \frac{F}{m} \end{aligned} \quad (7-17)$$

Diese Gleichungen werden in die Matrixschreibweise überführt und anschließend mit einer numerischen Methode gemäß Tabelle 7-2 gelöst. Für die Auswertung der Ergebnisse stehen umfangreiche Hilfsmittel zur grafischen Aufbereitung zur Verfügung. Es wird deutlich, dass bereits bei der Modellierung umfangreiches Wissen über das System erforderlich ist.

Programme, die zu dieser Klasse von Simulationsprogrammen gehören, sind z. B. Matlab, Mathematica, Mathcad, Maple usw. Sie unterscheiden sich z. B. durch numerische

Anhang

oder symbolische Lösungsverfahren. Das System Mathematica rechnet im Gegensatz zu Matlab standardmäßig symbolisch.

Im Folgenden wird näher auf das mathematisch orientierte Simulationsprogramm Matlab eingegangen, da es in dieser Arbeit für die Simulation eingesetzt wird. Matlab basiert auf den in der Programmiersprache Fortran geschriebenen numerischen Funktionsbibliotheken LINPACK (engl.: Linear Algebra Package), EISPACK (Eigensystem Package) und BLAS (engl.: Basic Linear Algebra Subprograms) [4]. Mithilfe dieser Bibliotheken ist es möglich, Probleme der linearen Algebra zu lösen. Alle Operationen, die in Matlab durchgeführt werden, fußen ursprünglich auf einer Datenstruktur: der Matrix. Dies schlägt sich auch im Namen Matlab (engl.: Matrix Laboratory) nieder. Für weitere Informationen zum Thema Matrizenberechnungen ist Quelle [184] zu empfehlen.

Matlab ist nicht nur ein numerisches Werkzeug, sondern eine eigene Programmiersprache. Es handelt sich um eine sogenannte Scriptsprache, diese benötigt einen Interpreter. Somit können alle Kommandos unmittelbar in der Kommandooberfläche ausgeführt werden, ohne vorher kompiliert zu werden.

Matlab verfügt über eine große Zahl an Toolboxes, die das Funktionsangebot erweitern. Es wird z. B. eine Toolbox für die symbolische Mathematik angeboten. Die wohl bekannteste Toolbox ist Simulink, sie eignet sich für die Signalverarbeitung und wird im folgenden Kapitel näher beschrieben.

7.3.2 Signalfussorientierte Simulationsprogramme

In diesem Kapitel werden die signalfussorientierten Simulationsprogramme beschrieben. Eine häufig verwendete Beschreibungsform dieser Programme ist die Darstellung mithilfe von Blockschaltbildern [26]. Hierfür wird das Verhalten des Systems in einer grafischen Darstellung erfasst, die aus den einzelnen Teilblöcken sowie den Signalfüssen zwischen den Blöcken besteht (vgl. Abbildung 7-7) [11].

Ausgehend von der Gleichung (7-16) wird die signalfussorientierte Modellierung des Feder-Masse-Systems exemplarisch dargestellt, um die Unterschiede zu den anderen Simulationsprogrammen zu verdeutlichen.

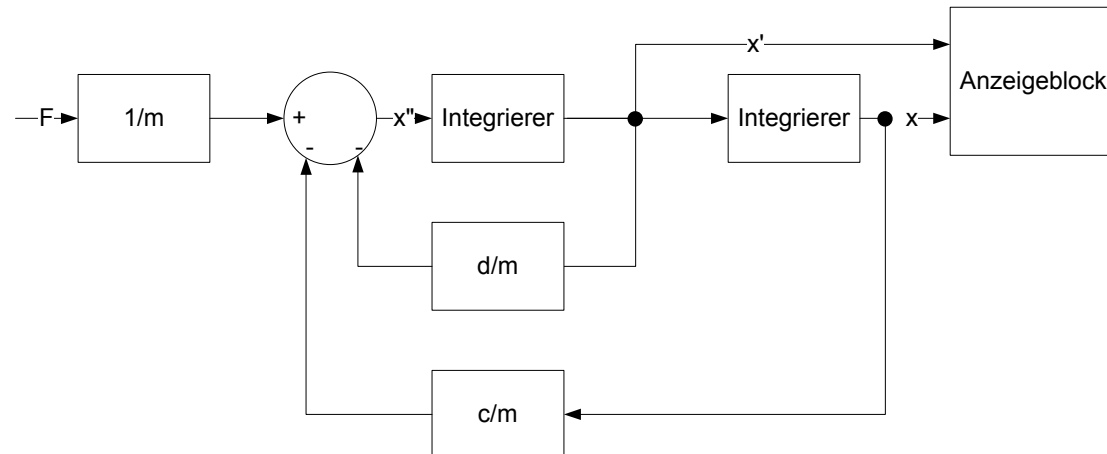


Abbildung 7-7 Signalfussorientiertes Modell eines Feder-Masse-Systems

Das in Abbildung 7-7 dargestellte Blockschaltbild mit den Signalfüssen entsteht, indem die nach \ddot{x} aufgelöste Gleichung (7-16) durch vorzeichenrichtige Summation der einzelnen Größen gebildet wird. Die höchste Ableitung wird so lange integriert und mit den jeweiligen Konstanten multipliziert, bis sich die gesuchte Funktion ergibt. Die benötigten Blöcke werden den Programmbibliotheken entnommen, verbunden und parametrisiert, für die Visualisierung stehen Anzeigeblocke zur Verfügung. Es wird deutlich, dass das Differenzialgleichungssystem für diese Art der Modellierung vom Benutzer aufgestellt werden muss, das kann für komplexe Systeme sehr unübersichtlich und umfangreich sein. Für regelungstechnische Anwendungen eignen sich diese Simulationsprogramme besonders gut, da die Blockschaltbild-darstellung identisch ist (vgl. [151]). Anhand des Simulationsprogramms Simulink wird im Folgenden die Funktionsweise eines signalfussorientierten Simulators exemplarisch beschrieben.

7.3 Systemsimulation

Bei dem Simulationswerkzeug Simulink handelt es sich um ein Programm zur Simulation dynamischer Systeme, das als Toolbox in die Matlab-Umgebung integriert ist. Die grafische Bedienoberfläche erlaubt die Darstellung des betrachteten Systems in der Form eines Blockdiagramms, bestehend aus Blöcken und Signalflüssen. In umfangreichen Blockbibliotheken stellt Simulink dem Benutzer vorgefertigte Funktionsblöcke zur Verfügung.

In Simulink werden also Blockschaltbilder benutzt, um die zeitabhängige Beziehung zwischen Signalflüssen- und Zustandsvariablen zu beschreiben. Dabei beschreiben die Signale die Flussgrößen zwischen den Blöcken zu jedem Simulationszeitpunkt. Die Beziehung der Zustandsgrößen und der Signalflussgrößen wird mithilfe der blockinternen Gleichungen beschrieben. Diese Gleichungen definieren die Abhängigkeiten zwischen den Ein- und Ausgangssignalen sowie den Zustandsvariablen. Das Ausgangssignal wird also mithilfe des Eingangssignals und der Blockgleichungen berechnet [163].

Um das Modell simulieren zu können, muss es zunächst kompiliert werden. Der wichtigste Schritt hierbei ist die Festlegung der Simulationsreihenfolge. Während der Simulation berechnet Simulink zu jedem Simulationszeitschritt die Zustands- und Ausgangsvariablen. Die Berechnungsreihenfolge der Blöcke ist dabei von entscheidender Bedeutung. Ein Block, der seine Ausgangsvariable in Abhängigkeit der Eingangsvariablen berechnet, muss nach dem Block, welcher die Eingangsvariable zur Verfügung stellt, berechnet werden (Blockreihenfolge: 1. Blöcke ohne Input, 2. zustandsbasierte Blöcke, 3. algebraische Blöcke). In Abbildung 7-8 ist die Blockreihenfolge durch rote Zahlen rechts oben neben den Blöcken dargestellt. Für die Sortierungsregeln wird an dieser Stelle auf die Quelle [163] verwiesen.

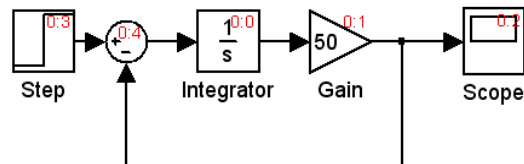


Abbildung 7-8 Beispiel einer Sortierungsreihenfolge für Simulink

Die Blöcke werden in der Simulation also sequenziell in der festgelegten Simulationsreihenfolge berechnet. Hier wird das Problem der algebraischen Schleifen deutlich, die im Kapitel 7.1 vorgestellt wurden. Durch das direkte Zurückführen von Ausgangsgrößen auf den Eingang können Konvergenzprobleme auftreten. Dies gilt für Blöcke, die keine Zustandsvariable besitzen (z. B. Gain, Sum, Product, usw.).

In der Simulationsphase berechnet Simulink sukzessiv die Zustands- und Ausgangsvariablen vom Simulationsstartzeitpunkt bis zum -endzeitpunkt. Die sukzessiven Zeitpunkte, zu denen die Variablen bestimmt werden, werden auch als Zeitschrittweite bezeichnet. Die Simulationsschleife für die Berechnung der einzelnen Zeitschritte sieht in Simulink folgendermaßen aus:

1. Berechnung der Blockausgangsgrößen
2. Berechnung der nächsten Zustände (zustandsbasierte Blöcke)
3. Prüfung von Diskontinuitäten
4. Zeitfortschritt
5. Durchlaufen von 1. bis 4., bis das Simulationsende erreicht ist

Für weitere Informationen zum Simulationsprogramm sei an dieser Stelle auf folgende Quellen verwiesen [4], [11], [163]. Im folgenden Kapitel werden die symbolorientierten Simulationsprogramme vorgestellt.

7.3.3 Symbolorientierte Simulationsprogramme

Die symbolorientierten Simulationsprogramme stellen eine grafische Modellierung des physikalischen Systems bereit. In anschaulicher Form entspricht das grafische Modell von der Symbolik her weitestgehend dem physikalischen Modell [26]. Die Verbindungen zwischen den Komponenten stellen den physikalischen Zusammenhang her.

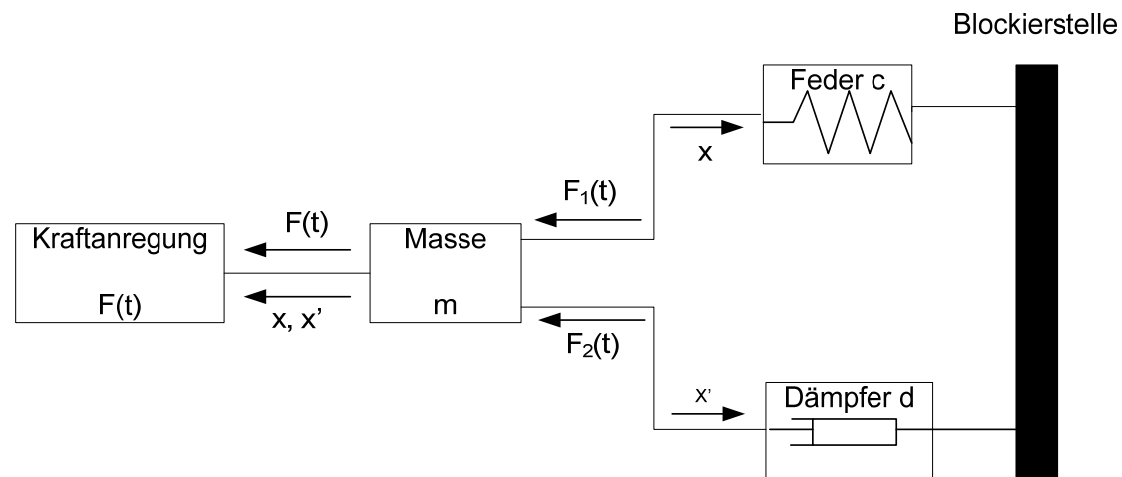


Abbildung 7-9 Symbolisches Modell eines Feder-Masse-Systems [21]

In Abbildung 7-9 ist das Feder-Masse-System aus Gleichung (7-16) als grafisches Modell eines symbolorientierten Simulationsprogramms dargestellt. Mithilfe der Verbindungen zwischen den Komponenten werden die Kräfte, Auslenkungen und Geschwindigkeiten übergeben. Die den Komponenten hinterlegten mathematisch-physikalischen Modelle bestimmen aus den Eingangsgrößen die Ausgangsgrößen [21]. Es wird deutlich, dass nicht die Differenzialgleichungen des Systems aufgestellt werden müssen, wie es bei den signalfussorientierten Programmen der Fall ist. Zur Modellierung des Systems müssen lediglich die Komponenten verschaltet werden, wobei die Einzelkomponenten aus Programmbibliotheken entnommen werden können. Das Simulationsprogramm übernimmt das Aufstellen der differenzialalgebraischen Gleichungen, womit sich der Modellierungsaufwand erheblich reduziert.

In einigen Quellen wird diese Art der Modellierung akausale Modellierung genannt [68], [85]. Bei der akausalen Modellierung sind die Ein- und Ausgänge nicht explizit definiert. Komponenten werden anhand ihrer physischen Verbindungen zu anderen und ihrer inneren mathematisch-physikalischen Zusammenhänge beschrieben. Durch das Verbinden der Komponenten werden Flüsse und Potenziale generiert (vgl. Tabelle 7-3). Das Simulationsprogramm transformiert das System der vorliegenden algebraischen und Differenzialgleichungen symbolisch in eine Form, die sich mit den bekannten Integrationsverfahren (vgl. Tabelle 7-2) integrieren lässt. Im Gegensatz dazu werden die bereits beschriebenen signalfussorientierten Simulationsprogramme als kausale Modellierung bezeichnet. Bei dieser Art der Modellierung (z. B. Blockschaltbilder und Signalflüsse) werden direkte Zusammenhänge zu den vorangegangenen Zustandsgrößen vorausgesetzt. Eine Ausgangsgröße kann nur über mathematische Zusammenhänge aus einer oder mehreren Eingangsgrößen berechnet werden [68].

Anhand des Simulationsprogramms Saber des Softwareherstellers Synopsys wird im Folgenden die Vorgehensweise eines symbolorientierten Simulationsprogramms zur Lösung eines differenzial-algebraischen Gleichungssystems exemplarisch erläutert.

Bei Saber handelt es sich um einen Netzwerksimulator, der ursprünglich zur Simulation elektrischer Schaltungen entwickelt wurde. Das Programm eignet sich zur Simulation von Hybrid-Systemen (vgl. Kapitel 2.1) und wurde auf Basis von Spice-Simulatoren entwickelt. Mithilfe des mathematischen Simulators ist es möglich, komplexe multidisziplinäre Systeme, die durch Netzwerke aus Modellen und ihren Interaktionen dargestellt werden, zu lösen [94]. Wie bereits erwähnt, werden Bauelemente für das jeweilige System in Bibliotheken zur Verfügung gestellt. Durch das symbolische Verschalten der einzelnen Bauelemente ergibt sich ein Simulationsmodell. Für die Modellierung physikalischer Systeme werden zwei Variablentypen benötigt, die sogenannten Fluss- und Potenzialvariablen. In Tabelle 7-3 sind diese Variablen für verschiedene physikalische Bereiche dargestellt.

7.3 Systemsimulation

Physikalischer Bereich	Flussvariable	Potenzialvariable
Elektrisch	Strom	Spannung
Rotatorisch	Drehmoment	Winkelgeschwindigkeit
Mechanisch	Kraft	Translatorische Geschwindigkeit
Magnetisch	Fluss	Magnetomotorische Kraft
Strömungsmechanisch	Flussrate	Druck
Thermisch	Wärmestrom	Temperatur
Licht	Lichtstrom	Beleuchtungsstärke

Tabelle 7-3 Fluss- und Potenzialvariablen für verschiedene physikalische Bereiche

Jedes Element ist durch seine Schnittstelle an den Systemgrenzen (Potenziale und Flüsse), sowie sein inneres Verhalten mithilfe von Erhaltungssätzen und Differenzialgleichungen beschrieben [68].

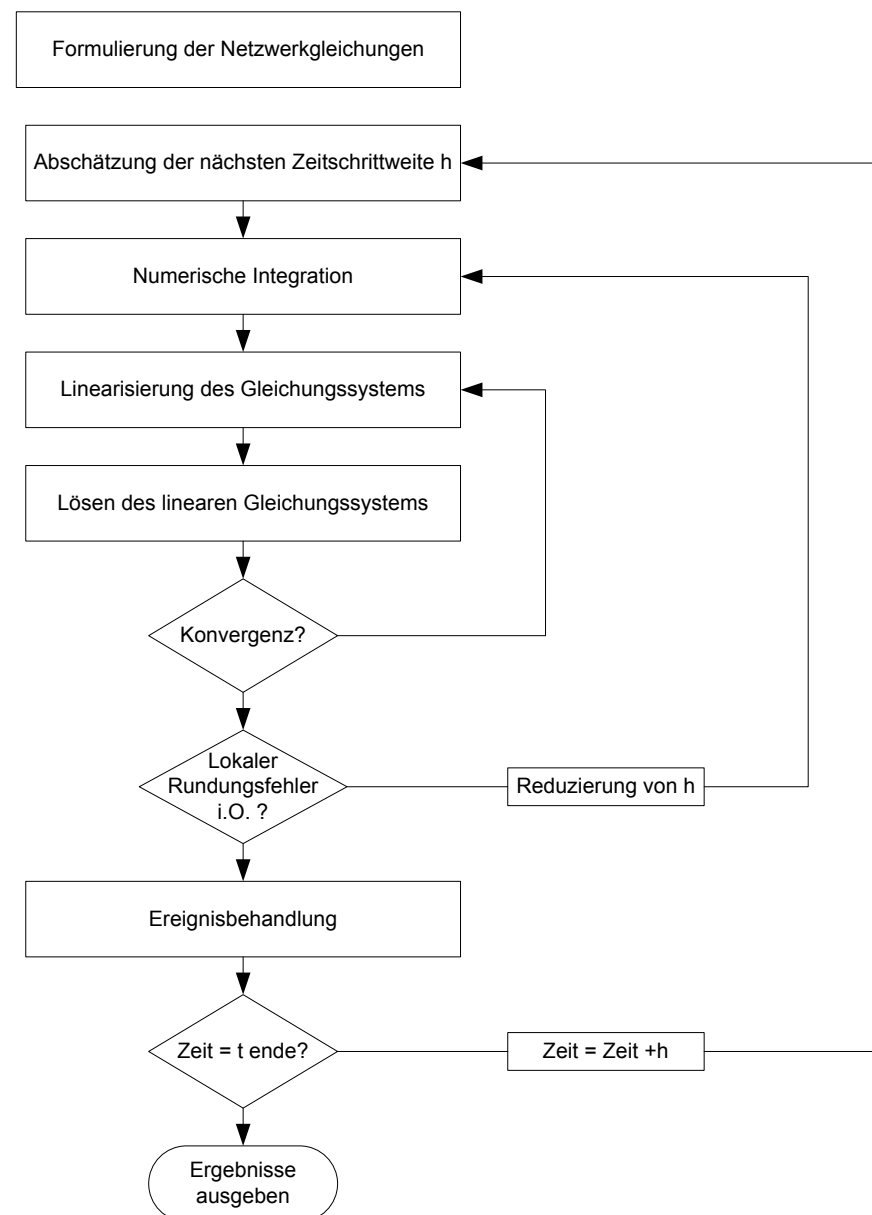


Abbildung 7-10 Simulation eines nichtlinearen differenzialalgebraischen Systems [32]

In Abbildung 7-10 ist der Simulationsprozess für die numerische Berechnung von Netzwerken dargestellt. Im Folgenden wird das Vorgehen zur Lösung eines nichtlinearen differenzialalgebraischen Gleichungssystems anhand eines elektrischen Systems beschrieben.

Anhang

Um die Ströme und Spannungen in einem elektrischen Netzwerk zu berechnen, muss als erster Schritt ein Gleichungssystem zur Beschreibung des Netzwerkes aufgestellt werden. Bei der Formulierung dieser Netzwerkgleichungen können unterschiedliche Methoden eingesetzt werden. Saper verwendet hier die kirchoffsche Knotenanalyse (Abk. KCL; engl.: Kirchhoff's Current Law) [37].

Nachdem die Netzwerkgleichungen aufgestellt wurden, muss das Gleichungssystem gelöst werden. Bei der Lösung dieses Gleichungssystems sind zwei Fälle zu unterscheiden. Die Lösung des Gleichungssystems an einem Gleichgewichtspunkt, auch DC-Analyse genannt, und die Lösung bei fortschreitender Zeit (transiente Analyse).

DC-Analyse

Das ursprüngliche elektrische Netzwerk wird derart verändert, dass alle Spulen durch Kurzschlüsse und alle Kondensatoren durch Unterbrechungen ersetzt werden, somit verschwinden alle Ableitungen nach der Zeit im Gleichungssystem. Das so entstehende Netzwerk repräsentiert ein lineares oder nichtlineares Gleichungssystem ohne Differenzialoperatoren. Die Lösung des Gleichungssystems dient als Initialisierungspunkt für die transiente Analyse.

Transiente Analyse

Zur Berechnung des transienten Verhaltens des Netzwerkes müssen die Netzwerkgleichungen bei kontinuierlich voranschreitender Zeit berechnet werden. Bei der transienten Analyse von elektrischen Netzwerken mit Kapazitäten und Induktivitäten treten Differenzialgleichungssysteme auf. In der Regel handelt es sich um steife schwach besetzte Differenzialgleichungssysteme. Der Simulationsprozess für die transiente Analyse ist in Abbildung 7-10 dargestellt.

Als Ansatz für die Lösung des nichtlinearen algebraischen Gleichungssystems dient das mehrdimensionale Newton-Verfahren [vgl. Gleichung (7-9)]. Basierend auf dieser Gleichung und der Schrittweite h kann der Funktionswert zum Zeitpunkt t ermittelt werden. Die im Gleichungssystem enthaltenen Ableitungen werden mithilfe der numerischen Integration (vgl. Tabelle 7-2) aufgelöst. Bei dem daraus resultierenden Gleichungssystem handelt es sich um ein nichtlineares Gleichungssystem, welches mithilfe des Newton-Verfahrens (vgl. Kapitel 7.1) linearisiert wird. Für die Lösung eines nichtlinearen Gleichungssystems wird hierfür die Jacobi-Matrix benötigt [vgl. Gleichung (7-8)]. Konvergiert das Verfahren gegen die Lösung und ist der Rundungsfehler in den erlaubten Grenzen, wird zum nächsten Zeitschritt vorangeschritten. Die Simulationsschleife wird so lange durchlaufen, bis das Ende der Simulation erreicht wird. Für weitere Informationen zur Lösung von DAE sei an dieser Stelle auf folgende Quelle verwiesen [32].

Basierend auf Erhaltungsgleichungen, ist es möglich, Gleichungssysteme für die in Tabelle 7-3 erwähnten physikalischen Bereiche in Analogie zu dem elektrischen Gleichungssystem aufzustellen [68]. In mechanischen Systemen sind z. B. die Summen aller internen Kräfte und Momente in einem Punkt gleich null. Früher wurden elektrische Ersatzmodelle für mechanische Systeme aufgebaut und dann auf Analogrechnern simuliert [26].

7.4 Datenaustauschformate und Informationsbereitstellung

In diesem Kapitel werden verschiedene Programme und Austauschformate analysiert, die für den Austausch von Simulationsdaten im Entwicklungsprozess eingesetzt werden können. Wie bereits in Kapitel 2.4.1 erwähnt wurde, ist die Standardisierung von Schnittstellen für den Austausch von Simulationsdaten und Modellen von besonderer Bedeutung. Es wird kein Anspruch auf Vollständigkeit erhoben, vielmehr werden die Vor- und Nachteile der bekanntesten und bereits eingesetzten Programme und Formate dargestellt.

7.4.1 Softwareprogramme zur Simulationsdatenverwaltung

Zunächst werden vier bereits in der Flugzeugindustrie eingesetzte Softwarelösungen für die Verwaltung von Simulationsdaten vorgestellt.

SimDMS

Das SimDMS (Simulationsdatenmanagementsystem) ist ein von EADS und Airbus entwickeltes Datenmanagementsystem zur Speicherung und Visualisierung von Simulations- und Modellierungsinformationen [122]. Das System ist betriebssystemunabhängig und durch eine Schnittstellendefinition an verschiedene Simulationswerkzeuge anpassbar. Es stellt eine einheitliche Plattform für die Ablage der Simulationsdaten zur Verfügung und erlaubt den interaktiven Zugriff auf Simulationsparameter. Als Basis für die Informationsvisualisierung dient eine gemeinsame Datenstruktur für die Ablage von Parametern aus gestaltorientierten Simulationen. Die vom SimDMS zu verwaltenden Informationen werden in die Kategorien Geometrie, Material, System und Prozess eingeteilt. Die definierten Kategorien sind die Basis für die Datenverwaltung und Informationsvisualisierung. Mithilfe einer einheitlichen Kennzeichnung der Daten in Informationsklassen, in Abhängigkeit von Qualität bzw. Reifegrad, erfolgt die Einstellung der Daten in das SimDMS. Für die Weiterverwendung ermöglicht diese Einteilung anderen Nutzern eine Einschätzung der Datenqualität. Eine einheitliche web-basierte Mensch-Maschine-Schnittstelle (Abk. GUI; engl.: Graphical User Interface) führt zudem zu einer interdisziplinären Kommunikation zwischen Prozessbeteiligten [122]. Das SimDMS eignet sich besonders für die Verwaltung von großen Datenmengen, wie sie in der gestaltorientierten Simulation vorhanden sind (z. B. CFD- und FEM-Simulationen).

SimManager

Der SimManager des Unternehmens MSC Software bietet die Möglichkeit, Simulationen in einer Datenbank zu verwalten. Die Daten werden dafür in eine Art Verzeichnisbaum einsortiert, dessen Struktur durch einen Administrator vorgegeben wird. Für mehrfach unter einem Namen abgelegte Daten werden Versionsnummern vergeben. Einzelne Mitarbeiter können auch eigene Versionsbäume, sogenannte Branches, implementieren und die Daten zu einem späteren Zeitpunkt wieder in den Hauptpfad einpflegen. Im SimManager können beliebige Dateien abgelegt werden (z. B. Dokumentationen, Simulationsmodelle, Ergebnisse usw.). Ein einheitlicher Datenbankzugriff wird mithilfe eines Web-Frontends bereitgestellt, welches zugleich den Einsatz auf nahezu jeder Plattform gewährleistet. Der SimManager ist nicht in der Lage, eine Transformation von Daten vorzunehmen, sodass es nicht vorgesehen ist, die Ausgangsdaten eines Simulators in die Datenbank einzupflegen, automatisch vom Server umwandeln und anschließend von einem beliebigen anderen Simulator in einem anderen Format auslesen zu lassen [150]. In Abbildung 7-11 ist der Datenfluss des SimManagers dargestellt. Bei dem eingesetzten Database Management System (Abk. DBMS) handelt es sich um DB2 von IBM. Das Web-Frontend wird über einen Apache Tomcat-Server zur Verfügung gestellt [17].

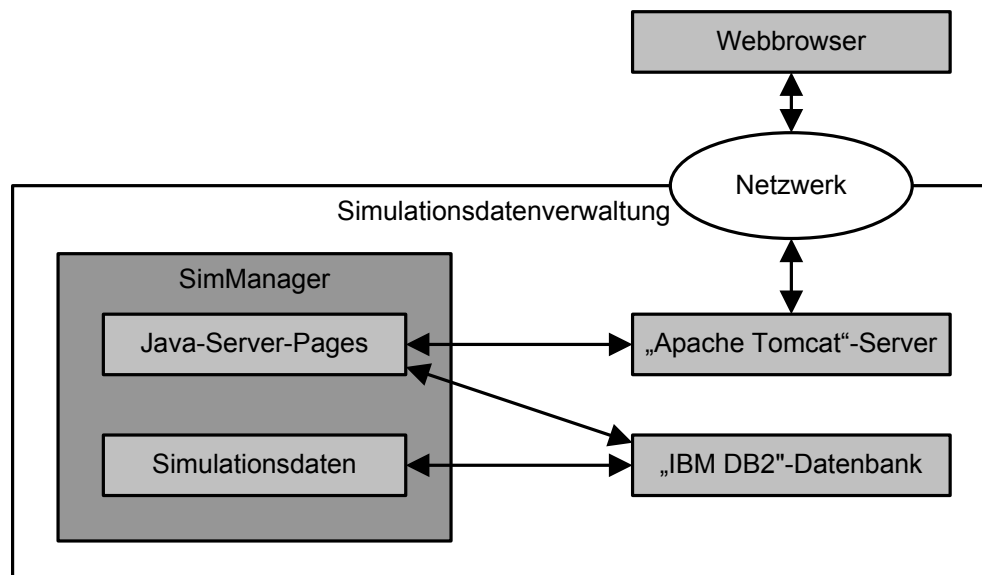


Abbildung 7-11 SimManager-Architektur [3]

In der Automobilindustrie konnte mithilfe des SimManagers die Entwicklungszeit um bis zu 25 % reduziert werden [17]. Zurzeit wird das Programm bei Audi und BMW unter dem Namen CAE-Bench und bei Airbus unter dem Namen AMDB eingesetzt. Zusammenfassend kann gesagt werden, dass der SimManager im Grunde ein Versionsverwaltungssystem ist, das auf Großunternehmen zugeschnitten ist.

Versionsverwaltungssysteme

Grundsätzlich eignen sich Versionsverwaltungssysteme wie das CVS (engl.: Concurrent Versions System) ebenfalls für die Verwaltung von Simulationsdaten. Die Vorteile des CVS sind der hohe Entwicklungsstand und die GPL-Lizenz (engl.: General Public License) [56], unter der die Anwendung veröffentlicht wurde. Das CVS-Projekt existiert seit 1989 und ist eine Weiterentwicklung des noch älteren Revision Control Systems. Der Einsatz von CVS ist seit Langem erprobt, wodurch seine Zuverlässigkeit gewährleistet ist. Da die Hauptzielgruppe Softwareentwickler sind, die ihre Quellcodes untereinander austauschen möchten, ist das CVS auf kleine Textdateien zugeschnitten. Die Ressourcen werden geschont, indem die Daten inkrementell (nur Änderungen) gespeichert werden. Das CVS ist für die Synchronisierung von Daten im lokalen Dateisystem mit Daten auf einem CVS-Server zuständig. Auf dem Server liegen die Daten in der Ablage (engl.: repositories), auf dieser Ablage ist es möglich, verschiedene Operationen durchzuführen. In Abbildung 7-12 ist der prinzipielle Aufbau einer Client-Server-Beziehung dargestellt.

Die Nachteile von CVS sind die Notwendigkeit der Installation der CVS-Software auf jedem Client und die geringe Geschwindigkeit bei der Verarbeitung großer Datenmengen. Da bei gestaltorientierten Simulationen (vgl. Kapitel 7.2) in der Regel sehr große Datenmengen anfallen, ist ein Versionsverwaltungssystem nicht für die Ablage von Simulationsdaten geeignet [150].

7.4 Datenaustauschformate und Informationsbereitstellung

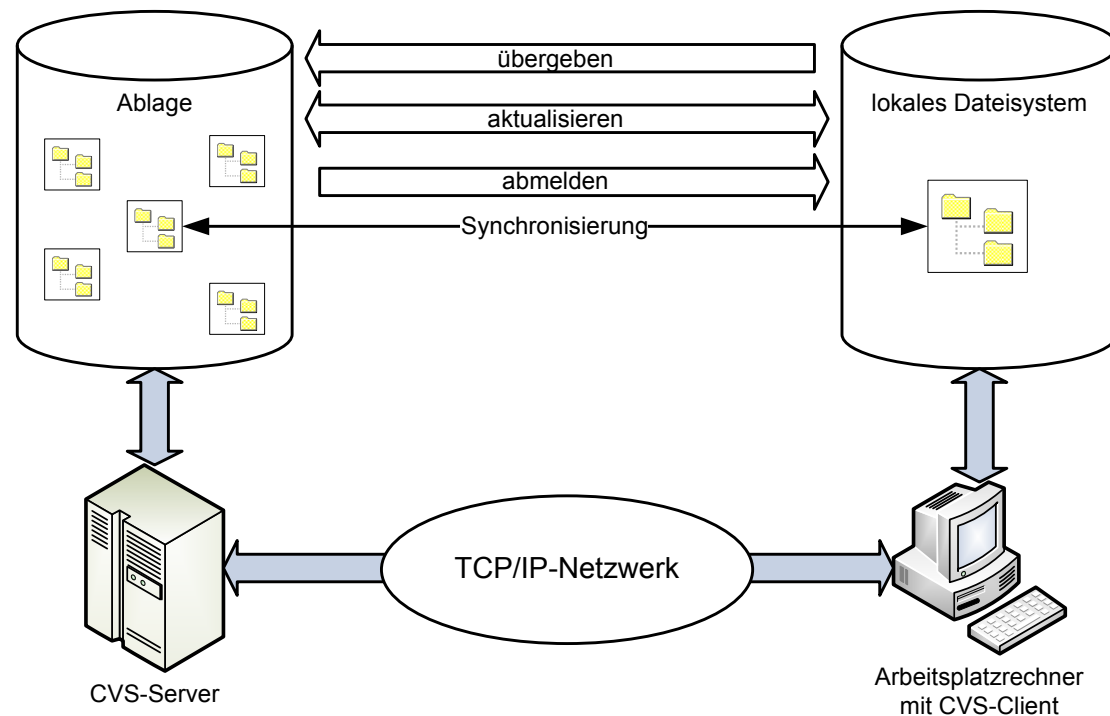


Abbildung 7-12 Funktionsweise von CVS

Documentum eRoom

Bei der Softwarelösung Documentum eRoom des Herstellers EMC handelt es sich um ein webbasiertes Dokumentenverwaltungssystem. Die Software wird hauptsächlich von Großunternehmen zur Verbesserung der standortübergreifenden Kommunikation eingesetzt. Neben der gleichzeitigen Bearbeitung eines Dokuments von mehreren Benutzern lässt sich eRoom als Versionsverwaltungssystem mit erweitertem Funktionsumfang umschreiben. Es bietet alle Funktionen, die auch von anderen Versionsverwaltungssystemen angeboten werden. Zusätzlich können dem Benutzer Informationen über die Änderungen an Dateien in regelmäßigen Abständen per Mail zugestellt werden. Nach der Anmeldung ist der Benutzer für alle anderen Nutzer sichtbar. In sogenannten Chat-Räumen kann die direkte Kommunikation zwischen den zurzeit angemeldeten Nutzern stattfinden. Da eRoom webbasiert ist, ist die Plattformunabhängigkeit der Client-Systeme sichergestellt. Zum Einloggen werden lediglich ein grafischer Webbrowser und eine JVM (engl.: Java Virtual Machine) benötigt. Grundsätzlich lässt sich eRoom auch für den Austausch von Simulationsdaten im Dateiformat nutzen. Allerdings gibt es kein Modul, mit dem die Daten aus eRoom direkt in eine Simulation importiert werden können [150], [46].

Neben den bisher beschriebenen professionellen Softwarelösungen kommen in der Praxis häufig wenig auf das Problem zugeschnittene, unprofessionelle und ineffiziente Verfahren zum Einsatz (vgl. Kapitel 3.2). Im folgenden Abschnitt werden die bekanntesten Datenaustauschformate dargestellt.

7.4.2 Verfügbare Simulationsdatenformate

Die vorgestellten Konzepte der Simulationsdatenverwaltung weisen eine gemeinsame Schwäche auf: Es ist nicht möglich, die Simulationsdaten in einem beliebigen Format abulegen und in einem beliebigen anderen Format auszulesen [150]. Im folgenden Abschnitt werden vorhandene Formate beschrieben, die für die herstellerunabhängige Speicherung von Simulationsdaten infrage kommen. In Kapitel 5.3.1 wird eine lose Kopplung von Simulationsprogrammen mithilfe eines standardisierten Austauschformats umgesetzt.

Hierarchical Data Format (HDF)

Bei dem Hierarchical Data Format handelt es sich um ein performantes, wenig speicherintensives und flexibles Binärdatenformat, welches vom NCSA (National Center for Supercomputing Applications) entwickelt wurde. Der Inhalt einer HDF-Datei ist eine hierarchisch angeordnete Ansammlung von Daten. In den Hierarchieebenen, den sogenannten Gruppen,

Anhang

können sich weitere Gruppen oder Datensätze befinden, wobei die Datensätze mehrdimensionale Felder, Tabellen, Bitmap-Grafiken oder Anmerkungen sein können. Jeder Gruppe sowie der Wurzel werden eindeutige Identifikatoren zugewiesen. In Abbildung 7-13 ist der Aufbau einer möglichen HDF-Datei dargestellt. Die Verknüpfungen zwischen den Elementen werden mithilfe von Zeigern, die auf einen Speicherbereich in der HDF-Datei zeigen, realisiert [162].

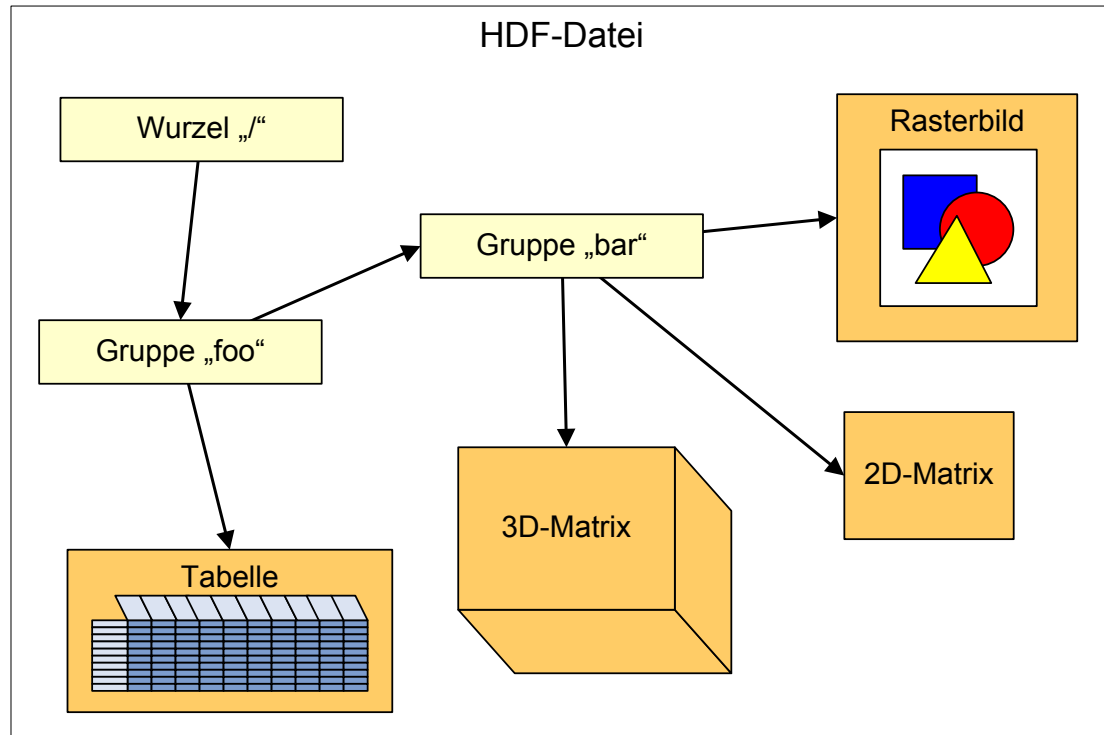


Abbildung 7-13 Aufbau einer HDF-Datei

Als Datentypen unterstützt das HDF unterschiedliche Gleitkomma-, Festkomma-, Zeichen- und Strukturdatentypen; für die Speicherplatzoptimierung stehen Kompressionsmethoden zur Verfügung. Die HDF-API (engl.: Application Programming Interface) ist in drei Schichten aufgeteilt: In der obersten Ebene existiert für jede Programmiersprache eine Schnittstelle, sodass in der Sprache auf Methoden, Funktionen, Felder und Variablen zugegriffen werden kann. Die Funktionalität dieser Schnittstelle verbirgt sich in der darunterliegenden Schicht und wird mithilfe von Bibliotheken bereitgestellt. Für gängige Plattformen existieren vorkompilierte Versionen, für alle weiteren Plattformen muss der Quellcode in eine entsprechende Bibliothek kompiliert werden. In der untersten Ebene werden für das Lesen und Schreiben von HDF-Dateien die Betriebssystemfunktionen für die Dateiein- und -ausgabe verwendet [150]. Zusammenfassend kann gesagt werden, dass es sich bei dem HDF um ein sehr ausgereiftes Format handelt, dem es lediglich an einer übersichtlichen API fehlt.

Extensible Markup Language (XML)

Die Extensible Markup Language (Abk. XML; engl. für erweiterbare Auszeichnungssprache) erlaubt die Beschreibung beliebiger Daten. Sie stellt einen Standard zur Modellierung von strukturierten Daten in Form einer Baumstruktur dar, der vom World Wide Web Consortium (Abk. W3C) definiert wird. In der XML sind Regeln für den Aufbau von Dokumenten definiert, die Daten enthalten [190]. Der Vorteil von XML gegenüber anderen Austauschformaten besteht darin, dass sogenannte Parser (engl. für Syntaxanalysator) in allen gängigen Programmiersprachen verfügbar sind und dass die Definition der Sprache von dem unabhängigen Gremium W3C durchgeführt wird.

Eine XML-API enthält außer dem Parser auch weitere Algorithmen zum Validieren von Inhalten (XML-Schemata und Dokumenttyp-Definition) und zur Transformation der Inhalte in andere Formate (Extensible Stylesheet Language). Der Nachteil von XML besteht darin, dass Daten im XML-Format deutlich mehr Speicher belegen, als es in den meisten anderen Formaten der Fall ist [150].

7.4 Datenaustauschformate und Informationsbereitstellung

Comma Separated Values (CSV)

Comma Separated Values (Abk. CSV; deutsch: durch Kommas getrennte Werte) ist ein universelles Format zum Speichern von Tabellen. Eine CSV-Datei setzt sich aus zwei Teilen zusammen. In der ersten Zeile der Tabelle befinden sich die Feldernamen. In den folgenden Zeilen stehen die dazugehörigen Werte, wobei die Felder einer Zeile durch ein Trennzeichen voneinander abgegrenzt werden. Pro Datei kann genau eine Tabelle abgelegt werden. Dieses Format eignet sich nicht für die Speicherung sämtlicher Simulationsdaten. Es können lediglich Simulationsdaten, die sich mit genau einer Matrix beschreiben lassen, mithilfe des CSV-Formats dargestellt werden [150].

XML Metadata Interchange (XMI)

Das XML Metadata Interexchange-Format ist entstanden, um eine Beschreibungssprache für die Modellierung von Software zu erschaffen. Es handelt sich hierbei um einen Standard der Object Management Group (Abk. OMG), der den Datenaustausch von Objekten auf Basis von Metamodellen nach der Meta Object Facility (Abk. MOF) gewährleistet [164]. Durch die Tatsache, dass XMI-Metadaten verwaltet, ist das XMI-Format lediglich für abstrakte Simulationsdaten geeignet, nicht aber für konkrete [150].

Ergänzend zu den Datenaustauschformaten für die Simulationsdaten existieren bereits individuelle Simulationsmodellaustauschformate, die in der Luftfahrtindustrie zum Einsatz kommen. Im Folgenden wird aus diesem Grund eine Richtlinie für den Austausch von Simulationsmodellen der Airbus Deutschland GmbH vorgestellt.

7.5 Prozessanalyse mit ARIS

Um die Defizite im derzeitigen Prozess zu identifizieren, wurde der Entwicklungsprozess der verschiedenen Fachabteilungen mit dem Softwareprogramm ARIS des Unternehmens IDS Scheer modelliert. ARIS eignet sich zur Modellierung und Verbesserung von Geschäftsprozessen. In diesem Kapitel ist als Beispiel ein Ausschnitt (Konzeptphase) der in ARIS modellierten Klimasystementwicklung dargestellt. Das folgende Bild zeigt die ARIS-Modellierungskonventionen:

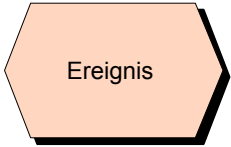

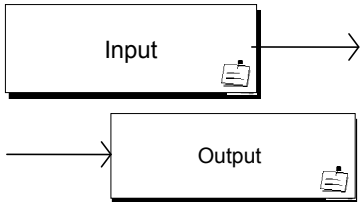
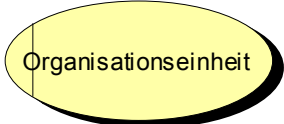
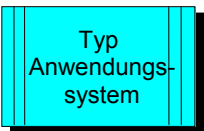
Objekttypen	Symbol	Beispiel/Erklärung
Ereignis		<ul style="list-style-type: none"> zum Beispiel „Detaillierungsphase beendet“
Funktion		<ul style="list-style-type: none"> zum Beispiel „Simulation durchführen“
Fachbegriff (Input/Output)		<ul style="list-style-type: none"> Informationen, Dokumente, Eingabeparameter, Ergebnisse, Simulationsmodelle usw. zum Beispiel „Physikalische Randbedingungen“, „Energieverbrauch“
Organisationseinheit		<ul style="list-style-type: none"> organisatorische Einheit, zum Beispiel „EYV“
Anwendungssystem		<ul style="list-style-type: none"> Simulationsprogramm, Datenbank, Datenmanagementsystem usw. zum Beispiel „Simulator“, „EDMS“ u. Ä.

Tabelle 7-4 Modellierungskonventionen ARIS

7.5 Prozessanalyse mit ARIS

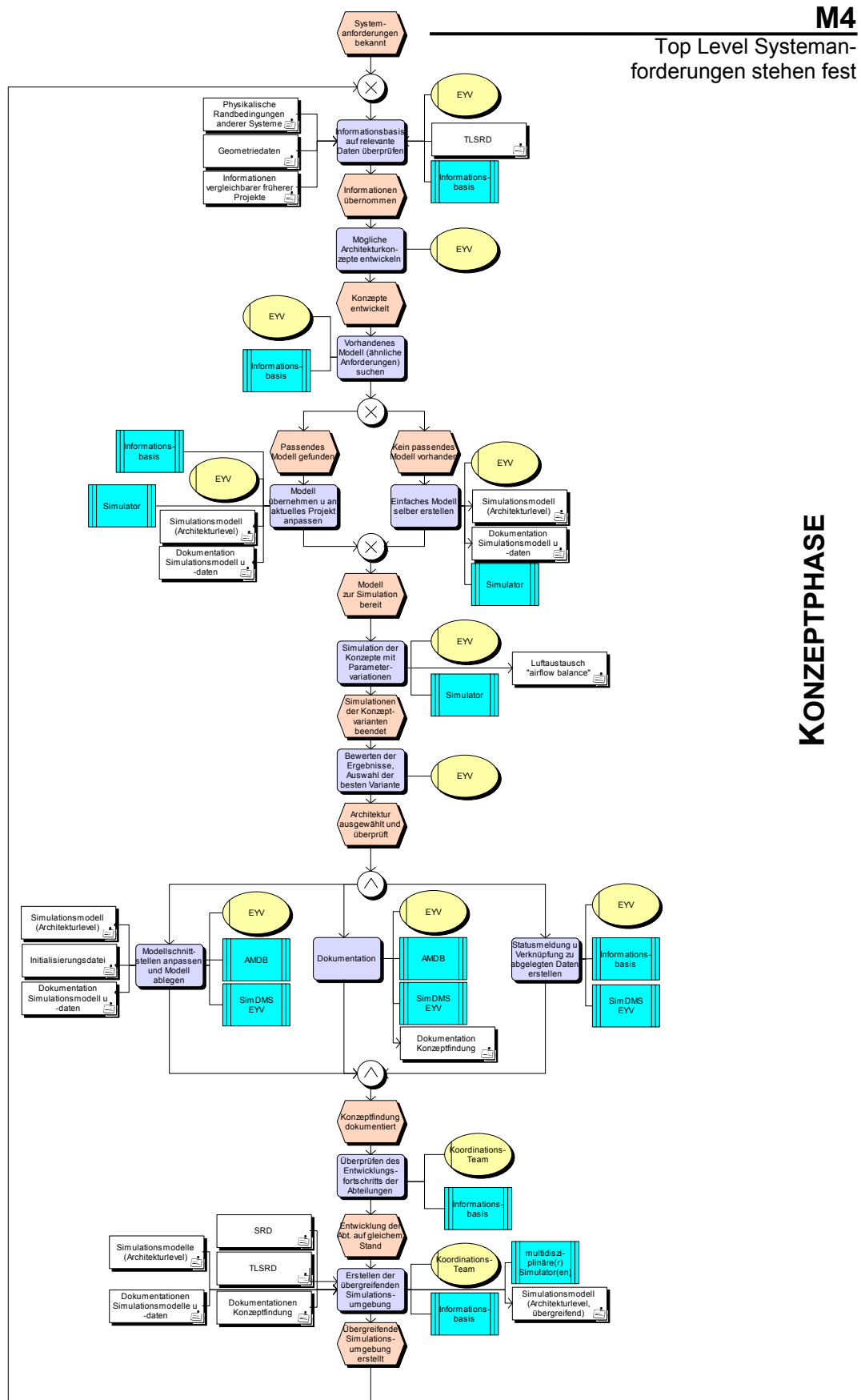
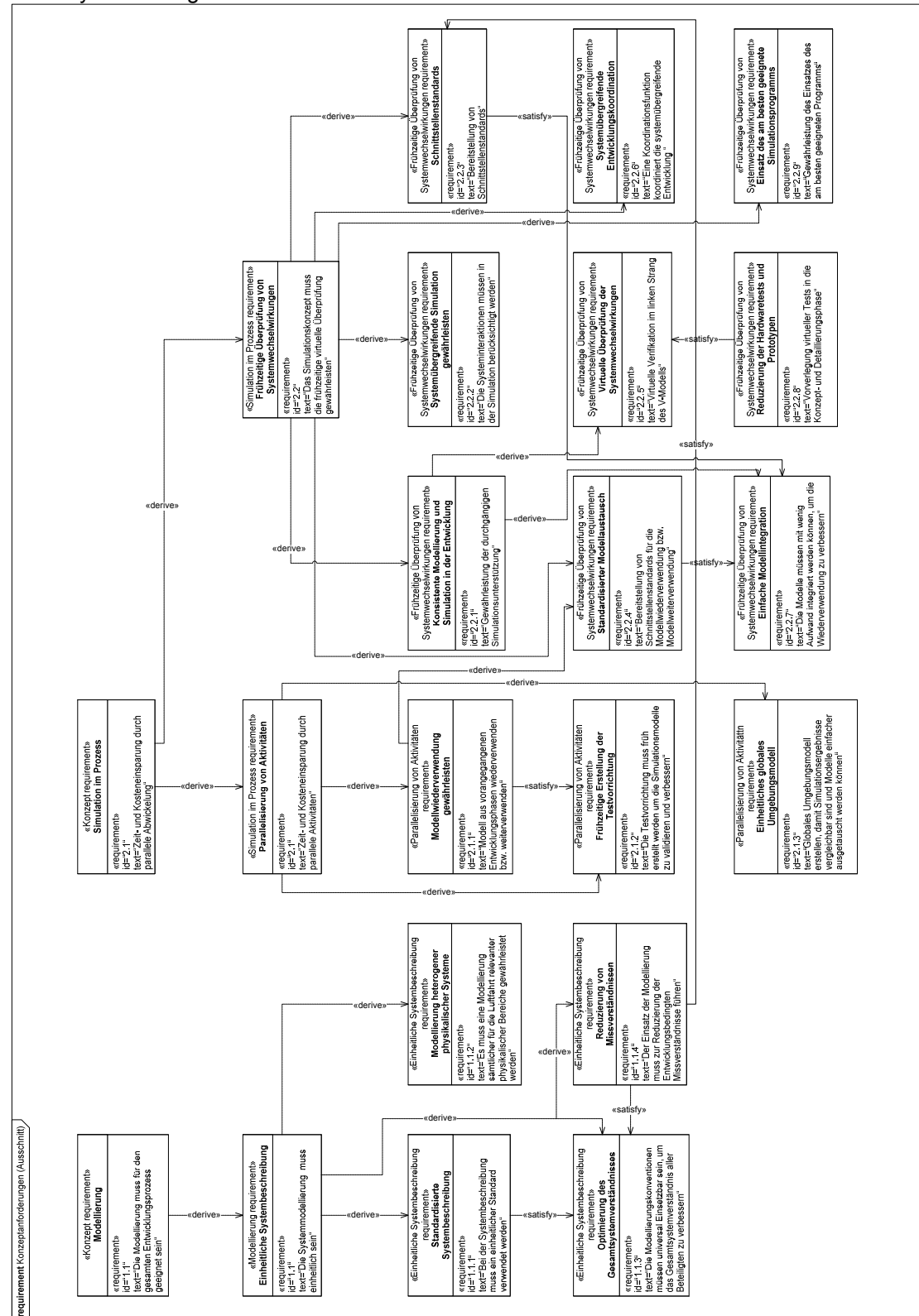


Abbildung 7-14 Konzeptanforderungen in SysML



7.6 Modellierung mit SysML

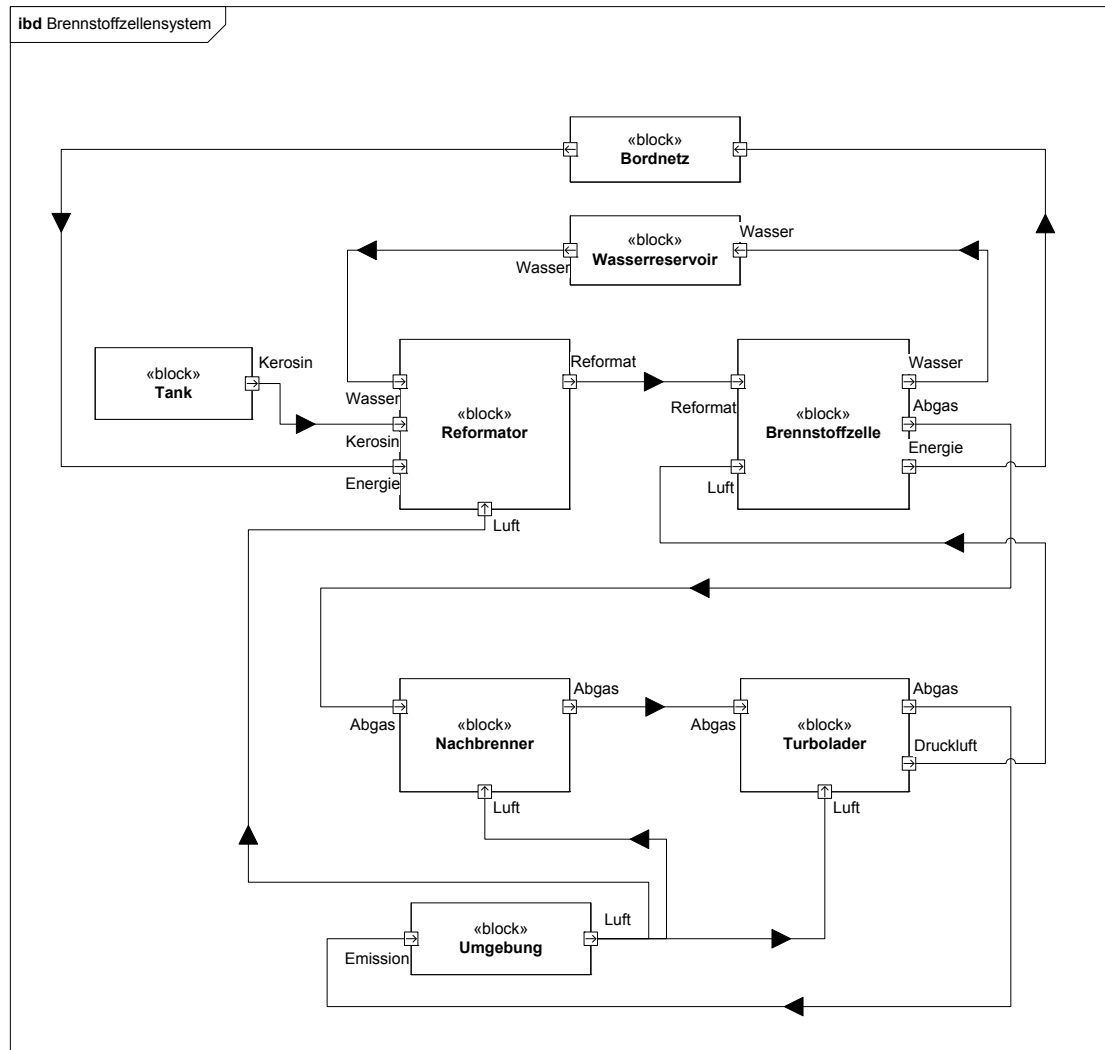


Abbildung 7-15 Blockschaftbild eines Brennstoffzellensystems in SysML

7.7 Standardisierte Schnittstelle

In Tabelle 7-5 ist als Beispiel für eine standardisierte Modellschnittstelle ein Vektor mit 17 Zeilen dargestellt. Dieser Vektor eignet sich für die Modellierung von Brennstoffzellensystemen. Der Einsatz des Vektors ermöglicht es, beliebige Stoffströme zwischen den Modellkomponenten im Brennstoffzellensystem zu berücksichtigen. Dieser standardisierte Flow Vector wird in der dargestellten Form in der FCLib des Unternehmens EUTech verwendet [48].

Position	Name	Symbol	Einheit
1	Molstrom	\dot{n}	mol/s
2	Temperatur	T	K
3	Druck	p	Pa
4	Enthalpiestrom	H	W
5	Entropiestrom	S	W/K
6	Dampfanteil	x	-
7	Gibbssche Energie	G	W
8	Wärmekapazität (p=konst.)	C_p	W/K
9	Relative Feuchtigkeit	Φ	-
10	-	-	-
11	molarer Anteil CH ₄	Ψ_{CH_4}	-
12	molarer Anteil H ₂ O	Ψ_{H_2O}	-
13	molarer Anteil CO	Ψ_{CO}	-
14	molarer Anteil CO ₂	Ψ_{CO_2}	-
15	molarer Anteil H ₂	Ψ_{H_2}	-
16	molarer Anteil N ₂	Ψ_{N_2}	-
17	molarer Anteil O ₂	Ψ_{O_2}	-

Tabelle 7-5 Standardisierte Modellschnittstelle

Literaturverzeichnis

- [1] Abst, D. Grundkurs JAVA: von den Grundlagen bis zu Datenbank- und Netzanwendungen
5., überarb. und erw. Auflage
Wiesbaden: Vieweg, 2008
ISBN: 3-8348-0417-7
- [2] Albers, A.;
Nowicki, L. Integration der Simulation in die Produktentwicklung
In: Tagungsband Abschlusskonferenz zum BMBF-Projekt ELAnO (Bremen)
Stuttgart: Fraunhofer IRB Verlag, 2003, S. 141–147
- [3] Anderl, R. Produktdatentechnologie A CAD-Systeme und CAX-Prozessketten
Teil 2
Skript zur Vorlesung, Fachgebiet Datenverarbeitung in der Konstruktion
Darmstadt: Technische Universität Darmstadt, 2003
- [4] Angermann, A. et al. Matlab-Simulink-Stateflow
Grundlagen, Toolboxes, Beispiele
2. Auflage
München: Oldenbourg, 2003
ISBN: 3-486-27377-9
- [5] Astles, S. Dimensional Management: The Reasons For Change
DCS – Dimensional Management Forum, April 2005
- [6] Balasubramanian, B. et al. Simulation im Automobilbau – von der Idee bis zum Kundenfahrzeug
In: VDI-Bericht 1215 Simulation in der Praxis – Neue Produkte effizienter Entwickeln, 1995, S. 1–17
ISBN: 3-18-091215-
- [7] Bärwolff, G. Numerik für Ingenieure, Physiker und Informatiker
Für Bachelor und Diplom
Berlin: Spektrum akademischer Verlag, 2007
ISBN: 3-8274-1689-2
- [8] Bathe, K.-J. Finite-Elemente-Methoden
2. Auflage
Berlin u. a.: Springer, 2002
ISBN: 3-540-66806-3
- [9] Bayer, J.;
Collisi, T. Simulation in der Automobilproduktion
Berlin: Springer, 2003
ISBN: 3-540-44192-1
- [10] Berners-Lee, T.; Fielding, R.; Frystyk, H. Hypertext Transfer Protocol – HTTP/1.0
RFC 1945, IETF, Mai 1996
[online]. Copyright: -, letzte Aktualisierung: 27.03.2008, erhältlich im Internet unter
<http://www.ietf.org/rfc/rfc1945.txt>
- [11] Beucher, O. Matlab Simulink Grundlagen Einführung
2. Auflage
Pearson Education Deutschland GmbH, 2002
ISBN: 3-8273-7042-6
- [12] Binet, M. Electrical Power Source and Aircraft Power Network
In: TEOS Forum
Paris: 2006
- [13] Blanchard, B. et al. Systems Engineering Vision 2020
Version 1.5
INCOSE, November 2005
- [14] Boeing Simulation Reference Markup Language: White Paper
In: Boeing SRML White Paper-Release version 8/15/2002
Boeing, 2002
- [15] Bossel, H. Systeme, Dynamik, Simulation
Modellbildung, Analyse und Simulation komplexer Systeme

Literaturverzeichnis

- Zierenberg: Books on Demand GmbH, 2004
ISBN: 3-8334-0984-3
- [16] Braess, H.-H.; Seiffert, U. Handbuch Kraftfahrzeugtechnik
5., überarb. und erw. Auflage
Wiesbaden: Vieweg, 2007
ISBN: 3-8348-0222-4
- [17] Brandner, B. MSC SimManager
[online]. Copyright: -, letzte Aktualisierung: 18.07.2007, erhältlich
im Internet unter
<http://www.mscsoftware.com/products/simmanager.cfm?Q=131&Z=288>
- [18] Bräunling, W. J. G. Flugzeugtriebwerke
2., erweiterte Auflage
Berlin: Springer Verlag, 2004
ISBN: 3-540-40589-5
- [19] Bräunling, W. J.G. Flugzeugtriebwerke : Grundlagen, Aero-Thermodynamik,
Kreisprozesse, thermische Turbomaschinen, Komponenten und
Emissionen
Ausgabe: 2
Berlin: Springer, 2004
ISBN: 3-540-40589-5
- [20] Bronstein I. N. et al. Taschenbuch der Mathematik
2. Auflage
Verlag Harri Deutsch, 1995
ISBN: 3-8171-2002-8
- [21] Brychat, P. et al. Technische Simulation
Wuppertal: Vogel Buchverlag, 2004
ISBN: 3-8023-1971-0
- [22] Bundesamt für Sicherheit in der Informationstechnik
M 2.124 Geeignete Auswahl einer Datenbank-Software
[online]. Copyright: -, letzte Aktualisierung 27.03.2008, erhältlich im
Internet unter
<http://www.bsi.bund.de/gshb/deutsch/m/m02124.htm>
- [23] Bundesministerium des Inneren
V-Modell XT
Version 1.2.0
Hrsg.: Bundesministerium des Inneren, 2004
[online]. Copyright: -, letzte Aktualisierung: 19.05.2006, erhältlich
im Internet unter
<http://www.v-modell-xt.de/>
- [24] Calaprice, A. Einstein sagt. Zitate, Einfälle, Gedanken.
9. Auflage
Piper: 1999
ISBN: 3-492-22805-4
- [25] Campbell, D. J. Revolutionary Propulsion and Power for 21st Century Aviation
Report
Cleveland Ohio: NASA Glenn Research Center, 2003
- [26] Cellier, F.E. Continuous System Modeling
Springer-Verlag, 1991
ISBN: 0-387-97502-0
- [27] Cerbe, G. et al. Technische Thermodynamik
Theoretische Grundlagen und praktische Anwendungen
14. Auflage
München u. a.: Hanser, 2005
ISBN: 3-446-40281-0
- [28] Chen, Peter P. Der Entity-Relationship-Ansatz zum logischen Systementwurf :
Datenbank- und Programmmentwurf
Mannheim: BI-Wiss.-Verl., 1991
ISBN: 3-411-15311-3
- [29] Coelho, D. R. The VHDL handbook/David R. Coelho
Boston [u. a.]: Kluwer Acad. Publ., 1989

Literaturverzeichnis

- [30] Concawe, EUCAR, JRC ISBN: 0-7923-9031-8
Well-to-Wheels analysis of future automotive fuels and powertrains in the European context, Overview of the Results
JEC WTW study Version 2c 03/2007
[online]. Copyright: -, letzte Aktualisierung: 17.11.2007, erhältlich im Internet unter
<http://ies.jrc.cec.eu.int/wtw.html>
- [31] Conrad, R.-P.; Schwab, J. Simulationgestützte Planung und Optimierung von Fertigungsprozessen und Produktionsanlagen
In: VDI-Bericht 1215 Simulation in der Praxis – Neue Produkte effizienter Entwickeln, 1995, S. 249–258
ISBN: 3-18-091215-
- [32] Cooper, R. S. The Designer's Guide to Analog & Mixed-Signal Modelingl Illustrated with VHDL-AMS and MAST
Synopsys, 2004
- [33] Cwielong, M. Konzeptstudie über den Einsatz von Brennstoffzellensystemen in der zivilen Luftfahrt
Kiel: Fachhochschule, 2007
- [34] Daimler-Chrysler Antriebe und Kraftstoffe mit eingebauter Zukunft
In: Hightech Report Faszination Forschung & Technik, Heft 1/2005, S. 18–23
- [35] Daimler-Chrysler Hybrid-Autos.Infor, F-Cell B-Klasse 2005
[online]. Copyright: -, letzte Aktualisierung: 18.11.2007, erhältlich im Internet unter
<http://hybrid-autos.info/Mercedes-F-Cell-2005.html>
- [36] Dais, S. Virtuelle Produkte – Basis für optimierte Produktentstehungsprozesse
In: 7. Automobiltechnische Konf. 2003, Virtual Product Creation 2003
Stuttgart: 30.06.–01.07.2003, Seite 1–8, Paper-Nr. 5
- [37] Demel, J. JANAP - Ein Programm zur Simulation von elektrischen Netzwerken
Fakultät für Elektrotechnik
Technischen Universität Wien
Wien: 1989, erhältlich im Internet unter
<http://www.iue.tuwien.ac.at/phd/demel/node4.html>
- [38] Department of Defense Defense Modeling and Simulation Office: High Level Architecture, Run-Time Infrastructure, RTI 1.3-Next Generation Programmer's Guide
Version 5
Alexandria: 2000
- [39] DMSO HLA High Level Architecture (HLA) [online]. Copyright: -, letzte Aktualisierung: 18.07.2007, erhältlich im Internet unter
<https://www.dmsomil/public/transition/hla>
- [40] DMSO RTI DMSO RTI Commercialization Announcement
[online]. Copyright: -, letzte Aktualisierung: 27.03.2008, erhältlich im Internet unter
<https://www.dmsomil/public/library/projects/hla/rti/announcement.php>
- [41] Döllner, G. Verkürzung der Produktentwicklungszeit durch den Einsatz von Simulations- und Rapid-Prototyping-Verfahren in der Motorenentwicklung
In: VDI-Berichte 1374
Düsseldorf: VDI-Verlag, 1997, S. 165–187
- [42] Dronka, S. Die Simulation gekoppelter Mehrkörper- und Hydraulik-Modelle mit Erweiterung für Echtzeitsimulation
Aachen: Shaker, 2004
- [43] Eelmann J. High Temperature fuel cells as substitution of the conventional APU in a commercial aircraft
In: DGLR-2003-183 Deutscher Luft- und Raumfahrt Kongress

Literaturverzeichnis

- [44] Ehrlenspiel, K. München: 2003
Integrierte Produktentwicklung : Denkabläufe, Methodeneinsatz, Zusammenarbeit
2. Auflage
München: Hanser, 2003
ISBN: 3-446-22119-0
- [45] Eigner, M.; Stelzer, R. Produktdatenmanagement-Systeme: ein Leitfaden für Product Development und Life-cycle-Management.
Berlin: Springer Verlag, 2001
- [46] EMC² Software EMC² Documentum eRoom
[online]. Copyright: -, letzte Aktualisierung: 18.07.2007, erhältlich im Internet unter
<http://software.emc.com>
- [47] Esterel Technologies SCADE The Standard for the Development of Safety-Critical Embedded Software in the Avionics Industry
[online]. Copyright: -, letzte Aktualisierung: 18.07.2007, erhältlich im Internet unter
<http://www.esterel-technologies.com/>
- [48] EUtech FClib, User Manual
Simulation toolbox for the Design and Development of Fuel Cell Systems for use with Matlab/Simulink
Version 3.2.4
- [49] Faaß, R. Liquid Hydrogen Filled Aircraft, Final Technical Report Publishable Version
Hamburg: Airbus Deutschland GmbH, 2003
- [50] Fluent General Modeling Specification
[online]. Copyright: 1995–2008 by Fluent, letzte Aktualisierung 27.03.2008, erhältlich im Internet unter
http://www.fluent.com/software/fluent/modeling_spec.htm
- [51] Ford, N. Time-Triggered Protocol in the Automotive Domain and Ist Potential Benefits for Aerospace pplications
In: Open Forum on Time-Triggered Technologies
Toulouse: 24.–25.04.2007
- [52] Friedenthal, S. The Unique Focus of Systems Engineering
In: Insight INCOSE The Use of Systems Engineering in Large-Scale Emergencies Vol. 8, Issue 3, July 2006, S.33–34
- [53] Fujimoto, R. M. Parallel and distributed simulation systems
New York: Wiley-Interscience, 2000
ISBN: 0-471-18383-0
- [54] Gausemeier, J.; Lückel, J. Entwicklungsumgebungen Mechatronik: Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme
Paderborn: HNI, Heinz Nixdorf Inst., 2000
- [55] Gebhardt, A. Rapid Prototyping
1. Auflage
München: Hanser, 2003
ISBN: 3-446-21259-0
- [56] GNU Project GNU General Public License
[online]. Copyright: 2007, letzte Aktualisierung: 19.07.2007, erhältlich im Internet unter
<http://www.gnu.org/licenses/gpl.html>
- [57] Gora, H. J. Einsatzfelder der Simulation in der Automobilindustrie
In: Simulation in der Automobilproduktion
Berlin: Springer, 2003, S. 17–27
ISBN: 3-540-44192-1
- [58] Grahs, T.; Othmer, C. Computation of acoustic sound sources: Parametric study concerning the aero acoustic properties of wing mirror geometries
In: VDI-Bericht 1967.1 Berechnung und Simulation im Fahrzeugbau, 2006, S. 23–36
ISBN: 3-18-091967-1
- [59] Greimer, M.; Co-Simulation, gekoppelte Simulation oder Simulatorkopplung?

Literaturverzeichnis

- [60] Krüger, T.;
Linsel, P.
Hairer, E. et
al. Ein Versuch der Begriffsvereinheitlichung
In: O+P Zeitschrift für Fluidtechnik, Nov.–Dez. 2006, S. 572–576
Solving Ordinary Differential Equations I
Nonstiff Problems
Second Edition
Berlin [u. a.]: Springer, 1993
ISBN: 3-540-56670-8
- [61] Hairer, E. et
al. Solving Ordinary Differential Equations II
Stiff and differential-algebraic problems
Second Edition
Berlin [u. a.] : Springer, 1996
ISBN: 3-540-60452-9
- [62] Hamilton
Sundstrand Staudruckturbine ERPS08 Hamilton Sundstrand
[online]. Copyright: 2007, letzte Aktualisierung: 18.11.2007,
erhältlich im Internet unter
<http://www.hamiltonsundstrand.com>
- [63] Hamilton
Sundstrand [online]. Copyright: 2006–2007, letzte Aktualisierung: 12.02.2009,
erhältlich im Internet unter
<http://www.hamiltonsundstrand.com/vgn-ext-templating-hs/v/index.jsp?vgnextoid=ae09afca90151110VgnVCM100000bb01000aRCRD>
- [64] Haun, J.;
Haun, R. Brennstoffzellen in der Raumfahrt
Ausarbeitungsskript 2005
Darmstadt: Fachhochschule, 2005
- [65] Heise Der perfekte Kraftstoff
[online]. Copyright: 2007, letzte Aktualisierung: 18.11.2007,
erhältlich im Internet unter
<http://www.heise.de/autos/artikel/s/3750/1>]
- [66] Herwig, H. Strömungsmechanik A-Z
Eine systematische Einordnung von Begriffen und Konzepten der
Strömungsmechanik
Wiesbaden: Vieweg, 2004
ISBN: 3-528-03974-4
- [67] Hoffmann, H.-
P. UML 2.0-Based Systems Engineering Using a Model-Driven
Development Approach
I-Logix Inc.
- [68] Hommel, M. Parallelisierte Simulationsprozesse für virtuelles Prototyping in der
Automobilindustrie
Fakultät Elektrotechnik und Informationstechnik
Technischen Universität Carolo – Wilhelmina zu Braunschweig
Braunschweig: 2006
- [69] Hoogers, G. Fuel Cell technology Handbook
Boca Raton: CRC Press, 2003
ISBN: 0-8493-0877-1
- [70] INCOSE Systems engineering Handbook
Version 2a
Technical Board International Council on Systems Engineering
INCOSE, Juni 2004
- [71] Jänker, P.;
Wolff, C. Vorrichtung zur Erzeugung von Wasserstoffgas durch
Dehydrogenerierung von Kohlenwasserstoff-Brennstoffen
Patentschrift D 10 2005 044 926 B3
Anmeldung: 20.09.2005
- [72] Jenkinson, L.
R.; Simpkin,
P. Civil Jet Aircraft Design
London: Arnold, 1999
ISBN: 0-340-74152-x
- [73] Junghanns, A Ausführliche Fehlersimulation zur Absicherung von robustheit und
Verfügbarkeit mechatronischer Systeme
DaimlerChrysler Forschung und Technologie.
ASIM-Fachgruppentreffen
München: 20.–21. Februar 2006

Literaturverzeichnis

- [74] Kallmeyer, J. Wissensmanagement im Entwicklungsprozess der Flugzeugsysteme – Grundlagen und Anwendungen
Hamburg: HAW, 2000
- [75] Kamke, E. Differentialgleichungen: Lösungsmethoden und Lösungen
I Gewöhnliche Differentialgleichungen
9. Auflage
Stuttgart: Teubner, 1977
ISBN: 3-519-02017-3
- [76] Kamke, E. Differentialgleichungen: Lösungsmethoden und Lösungen
II Partielle Differentialgleichungen erster Ordnung für eine gesuchte Funktion
6. Auflage
Stuttgart: Teubner, 1979
ISBN: 3-519-02024-6
- [77] Kasper, R.; Koch, W.; Kayser, A.; Wolf, A. Integrierte Entwicklungsumgebung mechatronischer Kfz-Komponenten und Kfz-Systeme
In: VDI-Berichte 1374 Systemengineering in der KFZ-Entwicklung, Düsseldorf: VDI-Verlag 1997, S. 451–465
ISBN: 3-18-091374-6
- [78] Kersken, H.-P. et al. TENT – eine Softwareintegrationsumgebung für multidisziplinäre Simulationen.
In: Tagungsband Deutscher Luft- und Raumfahrtkongress 2002
- [79] Klein, B. FEM
Grundlagen und Anwendungen
Braunschweig: Vieweg, 1990
ISBN: 3-528-05125-6
- [80] Koegst, M.; Lantzsch, G.; Schneider, A. Wiederverwendung bei der verteilten Simulation durch Anwendung der High Level Architecture
Dresden: EAS, 2000
SFB 358-B1/D4-1/2000
- [81] Kohl, W. Systemsimulation im Entwicklungsprozess aus der Sicht von Systems Engineering. Fachbereich Elektrotechnik und Informationstechnik, Systems Engineering
ASIM-Fachgruppentreffen
München: 20.–21. Februar 2006
- [82] Korn, W. Kersoene Reforming in Order to Generate Hydrogen for Feeding Fuel Cell Stack on Helicopter
Anhalt: Hochschule für angewandte Wissenschaften, 2003
- [83] Körner, M.; Dreyer, M.; Möller, T. Entwicklung einer auf der Prandtl-Analogie basierenden Berechnungsmethode zur effizienten thermischen Analyse von PWK-Scheibenbremsen
In: VDI-Berichte 1967.1
Düsseldorf: VDI-Berichte, 2006, S. 37–60
- [84] Kosturiak, J.; Milan, G. Simulation von Produktionssystemen
Wien: Springer, 1995
ISBN: 3-211-82701-3
- [85] Kottmann, M. P. Komponentenorientierte Modellierung und Simulation kombinierter Systeme
Dissertation
Eidgenössische Technische Hochschule Zürich, 1997
ETH-No. 12294
- [86] Krause, F.-L.; Franke, H.-J. Innovationspotential in der Produktentwicklung
München: Hanser, 2007
ISBN: 3-446-40667-0
- [87] Krause, F.-L.; Tang, T. Integrierte Virtuelle Produktentstehung (iViP) Abschlussbericht.
Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik
Berlin Stuttgart: Fraunhofer IRB Verlag, 2002
- [88] Krauss, P. Softwareentwicklung für mechatronische Applikationen
In: VDI-Bericht 1842 Mechatronischer Systementwurf

Literaturverzeichnis

- VDI Verlag, 2004, S. 127–132
ISBN: 3-18-091842-x
- [89] Kuhl F.,
Weatherly R.,
Dahmann J. Creating Computer Simulation Systems: An Introduction to the
High Level Architecture
Upper Saddle River: Prentice Hall PTR, 2000
ISBN: 0-13-022511-8
- [90] Kurzweil, P. Brennstoffzellentechnik Grundlagen, Komponenten, Systeme,
Anwendungen
1. Auflage
Wiesbaden: Vieweg, 2003
- [91] Kvasnicka, P. Durchgängige Simulationsumgebung zur Entwicklung und
Absicherung von fahrdynamischen Regelsystemen
In: VDI-Bericht 1967.1 Berechnung und Simulation im
Fahrzeugbau, 2006, S. 387–404
ISBN: 3-18-091967-1
- [92] Lahl, U. Clean Moves Perspectives on Mobility
In: Expo Hannover 16–20 April 2007
Hannover: Umweltministerium, 2007
[online]. Copyright: -, letzte Aktualisierung: 17.11.2007, erhältlich
im Internet unter
<http://www.bmu.de/verkehr/downloads/doc/39164.php>
- [93] Langermann,
R Unterstützung des Entwicklungsprozesses eines Brennstoff-
zellensystems an Bord eines konventionellen Flugzeugs
In: Jahrbuch der DGLR 2005 Band I + II
Friedrichshafen: Deutscher Luft- und Raumfahrtkongress, 2005
- [94] Langermann,
R. Spezifikation einer Simulationsumgebung für die Entwicklung eines
Brennstoffzellen-APU-Demonstrators
Diplomarbeit F0403-D
Institut für Luft- und Raumfahrtssysteme
Technischen Universität Carolo – Wilhelmina zu Braunschweig
Braunschweig: 2004
- [95] Langermann,
R. Spezifikation einer Simulationsumgebung für die Entwicklung eines
Brennstoffzellen-APU-Demonstrators
René Langermann
Braunschweig: TU-Braunschweig ILR, April 2004
Dokumentennummer: F0403-D
- [96] Langermann,
R.; Ullmann,
M.; Vörsmann,
P. Installation analysis of a fuel cell system on board an aircraft
In: International Aerospace CFD Conference
Paris: 18.–19.06.2007
- [97] Langermann,
R.; Vörsmann,
P. Kontinuierliche Systemsimulationsunterstützung im
Flugzeugentwicklungsprozess anhand eines
Brennstoffzellensystems
In: Deutscher Luft- und Raumfahrt Kongress 2006
Braunschweig: DGLR, 2006
- [98] Lapena Rey,
N. The Fuel Cell Demonstrator Airplane
In: 10th Grove Fuel Cell Symposium, 25.–27. September 2007
London: Elsevier, 2007
[online]. Copyright: -, letzte Aktualisierung: 17.11.2007, erhältlich
im Internet unter
http://www.boeing.com/phantom/news/2007/q1/070327e_nr.html
- [99] Larminie, J.;
Dicks, A. Fuel Cell Systems Explained
Second Edition
West Sussex: Wiley, 2003
ISBN: 0-470-84857-x
- [100] Larsson, T. CFD – Shaping the Future of Formula One
In: Fluent Forum 2006
Bad Nauheim: 13.–14.11.2006, S. 1-2
- [101] Ledjeff-Hey,
K.; Brennstoffzellen
2. Auflage

Literaturverzeichnis

- [102] Mahlendorf, F.; Roes, J. Liang, A. Heidelberg: C. F. Müller, 2001
ISBN: 3-7880-7629-1
Emerging Fuel Cell Development at NASA for Aircraft Application
Lewis Field: Glenn Research Center, 2003
[online]. Copyright: -, letzte Aktualisierung: 17.11.2007, erhältlich
im Internet unter
<http://www.netl.doe.gov/publications/proceedings/03/seca/Anita%20Liang.pdf>
- [103] MÄK Technologies Homepage der MÄK Technologies
[online]. Copyright: -, letzte Aktualisierung: 27.03.2008, erhältlich
im Internet unter
<http://www.mak.com/>
- [104] Mathworks Mathworks
[online]. Copyright: 1994–2008 The MathWorks, letzte
Aktualisierung: 27.03.2008, erhältlich im Internet unter
<http://www.mathworks.de/products/matlab>
- [105] Maume, C. Systemanalyse und Simulation eines Brennstoffzellen-Hybrid-
Fahrzeugs mit autothermer Methanolreformierung
München, Technische Universität, 2002
- [106] Mehl, H. Methoden verteilter Simulation
Braunschweig u. a.: Vieweg, 1994
ISBN: 3-528-05439-5
- [107] Melchinger, A.; Schmitz, E.-U. Schneller und effizienter in der Entwicklung durch Simulation und
Optimierung
Symposium „Simulation in der Produkt- und Prozessentwicklung“,
5.–7.11.2003, Bremen
Stuttgart: Fraunhofer IRB Verlag
- [108] Merkt, T.; Krastel, M. Virtuelle Produktabsicherung auf Basis eines CAE-
Datenmanagements
In: VDI-Bericht 1967.1 Berechnung und Simulation im
Fahrzeugbau, 2006, S. 245–258
ISBN: 3-18-091967-1
- [109] Merzinger, G. et al. Repetitorium der höheren Mathematik
4. Auflage
Binomi Verlag, 1999
ISBN: 3-923923-33-3
- [110] Metacrawler.de OHG MetaCrawler
[online]. Copyright: 2005, letzte Aktualisierung: 18.11.2007,
erhältlich im Internet unter
<http://www.metacrawler.de/>
- [111] Meyberg, K. et al. Höhere Mathematik 1
Differential- und Integralrechnung, Vektor- und Matrizenrechnung
3. Auflage
München: Springer-Verlag, 1995
ISBN: 3-540-59188-5
- [112] Milne, D. Modular Aerospace Controls (MAC) and Digital Engine Control
Applications
In: Open Forum on Time-Triggered Technologies
Toulouse: 24.–27. April 2004
- [113] Mohrdieck, C. Alternative Drive Train for Sustainable Mobility
In: Hybrid Vehicles and Energy Management 4th Symposium
Braunschweig: 14.–15. Februar 2007
Braunschweig: GZVB, 2007
ISBN: 978-3-937655-10-9
- [114] Moir, I.; Seabridge, A. Design and Development of Aircraft Systems an Introduction
AIAA Education Series, 2004
ISBN: 1-56347-722-x
- [115] Möller, D. Modellbildung, Simulation und Identifikation dynamischer Systeme
Lübeck: Springer-Verlag, 1992
ISBN: 3-540-55155-7

Literaturverzeichnis

- [116] Morris, A. Distributes MDO: The Way of the Future
In: CEAS Conference on Multidisciplinary Aircraft Design and Optomization
Köln: 25.–27.06.2001, S. 3–19
- [117] Navy Modeling and Simulation Management Office
Modeling and Simulation Verification, Validation and Accreditation Implementation Handbook
Volume 1 VV&A Framework, März 2004
[online]. Copyright: -, letzte Aktualisierung: 17.11.2007, erhältlich im Internet unter
http://www.simval.org/document/modeling_simulation_verification_validation_accreditation_implementation_handbook/081804_hdbk_Vol_1_final_033004.pdf
- [118] Neidhold, T.; Blochwitz, T.; Schreiber, U. et al.
Kooperation von Simulatoren durch Kopplung
In: 8. GMM-Workshop „Methoden und Werkzeuge zum Entwurf von Mikrosystemen“
Berlin: 2.–3.12.1999, S. 175–182
[online]. Copyright: -, letzte Aktualisierung: 13.11.2007, erhältlich im Internet unter
http://www.eas.iis.fhg.de/publications/papers/1999/035/index_de.html
- [119] Newport Aeronautical Sales
The Aerospace Technical Library ATA Chapter/Subsystem List
[online]. Copyright: 2001–2006, letzte Aktualisierung 18.07.2007, erhältlich im Internet unter
http://www.newportaero.com/home/ATA_CHAPTER
- [120] Novell, Inc.
Novelle Directory vs. Microsoft Active Directory Competitive White Paper
Waltham: Novell Inc., 2004
[online]. Copyright: -, letzte Aktualisierung: 17.11.2007, erhältlich im Internet unter
www.novell.com
- [121] Oertel, H. et al.
Strömungsmechanik
Grundlagen, Grundgleichungen, Lösungsmethoden, Softwarebeispiele
3. Auflage
Karlsruhe: Vieweg, 2004
ISBN: 3-528-23893-3
- [122] Olbert, M.
Methode zur interaktiven Informationsvisualisierung für virtuelle Prototypen in der Entwicklung von Flugzeugkabinen
Dissertation
Stuttgart: Fakultät für Luft- und Raumfahrttechnik und Geodäsie der Universität Stuttgart, 2006
- [123] Padur, V.; Schreiber, A.
TENT – eine Integrations- und Testumgebung für Simulation und Visualisierung
Köln: DLR SISTEC
[online]. Copyright: -, letzte Aktualisierung: 13.11.2007, erhältlich im Internet unter
<http://www.dlr.de/sc/>
- [124] Pahl, G.; Beitz, W.
Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung
5. Auflage
Berlin: Springer, 2003
ISBN: 3-540-00319-3
- [125] Paredis, C.; Kobi, R.
SysML-based Reference Models for Fluid Power Components
[online]. Copyright: -, letzte Aktualisierung: 16.07.07, erhältlich im Internet unter
<http://www.pslm.gatech.edu/topics/sysml/>
- [126] Parker
Air Separation Module (AMS) Parker
[online]. Copyright: 2007, letzte Aktualisierung 18.11.2007, erhältlich im Internet unter
<http://www.parker.com/ead/cm2.asp?cmid=3010&>

Literaturverzeichnis

- [127] Paschedag, A.R. CFD in der Verfahrenstechnik
Allgemeine Grundlagen und mehrphasige Anwendungen
Weinheim: Wiley-VCH, 2004
ISBN: 3-527-30994-2
- [128] Pemeas Celtec-P1000 MEA
[online]. Copyright: 2007, letzte Aktualisierung: 18.11.2007, erhältlich im Internet unter
<http://www2.pemeas.de/celtec.asp>
- [129] Pensky, H. Parallele und verteilte Simulation industrieller Produktionsprozesse
Fortschritt-Bericht VDI, Reihe 20, Rechnerunterstützte Verfahren, Nr. 376
Düsseldorf: VDI-Verlag, 2004
ISBN: 3-18-337620-2
- [130] Pitch Homepage der Pitch Technologie AB
[online]. Copyright: -, letzte Aktualisierung: 27.03.2008, erhältlich im Internet unter
<http://www.pitch.se/>
- [131] Porsche AG Lohner-Porsche 1900
[online]. Copyright: 2007, letzte Aktualisierung: 18.11.2007, erhältlich im Internet unter
www.porsche.de
- [132] Prähofer, H. System Theoretic Foundations for Combined Discret-Continuous System Simulation
Wien: Verb. d. Wiss. Ges. Österreichs, 1992
ISBN: 0259-0689
- [133] Puntigam, W.; Lang, G. et al. Instationäre Simulation des thermischen Verhaltens von Fahrzeugen durch Kopplung von Teilmodellen am Beispiel des Motorwärmeverhaltens
In: VDI-Berichte 1967.2
Düsseldorf: VDI-Berichte, 2006, S. 963–1008
- [134] Reichenberge C. Konzepte und Verfahren für die Software-Versionsverwaltung
Linz: Trauner, 1994
ISBN: 3-85320-673-5
- [135] Reichenthal, S. W. On the integration and distribution of XML-based simulations
Boeing
- [136] Reichenthal, S. W. On the integration and distribution of XML-based simulation
Boeing, 2002
Kontakt: steven.w.reichenthal@boeing.com
- [137] Reinhardt, A. Geschichten zur Simulation in der Automobilindustrie
In: Simulation in der Automobilproduktion, S. 7–15
Berlin: Springer, 2003
ISBN: 3-540-44192-1
- [138] Richter, S. Erstellung einer verteilten Simulation eines Brennstoffzellendemonstrators mit Hilfe der High Level Architecture
Braunschweig: TU-Braunschweig ILR, 2006
F 0523-D
- [139] Riel, A.; Hasewend, W.; Bogner, E.; Fischer, R. Modellbildung von Fahrzeug und Antriebsstrang im gesamten Entwicklungsprozess
In: Automobiltechnische Zeitschrift - ATZ, Band 106, Heft 6, 2004, S. 522–531
- [140] Robl, C. et al. Modell-basierte Entwicklung eines Maut-Erfassungssystems
In: Matlab Select, 2004, Heft 4, S. 9–13
- [141] Rothhammer, W. Airbus Fuel Cell System Approach
EEVR – Ref. PR0507283 – Issue 1
In: H₂Expo, Symposium
Hamburg: 31.08.05
- [142] Rulka, W. et al. Echtzeitsimulation der Fahrdynamik von Schienenfahrzeugen zur Inbetriebnahme der Fahrzeugsoftware im Labor-HiL

Literaturverzeichnis

- In: VDI-Bericht 1967.1 Berechnung und Simulation im Fahrzeugbau, 2006, S. 149–174
ISBN: 3-18-091967-1
- [143] Runge, W.; Gierer, G.; Deiss, H. Neue Wege in der Entwicklungspartnerschaft zwischen Fahrzeughersteller, Zulieferer und Elektroniklieferant
In: VDI-Berichte 1374
Düsseldorf: VDI-Verlag, 1997, S. 69–87
- [144] Sargent, R.G. Verification and Validation of Simulation Models
In: Proceedings of the 2003 Winter Simulation Conference
Syracuse, Department of Electrical Engineering and Computer Science, 2003
- [145] Saul, M.; Dick, J. Die Fahrzeugtür in der Simulation
In: VDI-Bericht 1967.1 Berechnung und Simulation im Fahrzeugbau, 2006, S. 1–12
ISBN: 3-18-091967-1
- [146] Schäfer, K. Entwicklung mechatronischer Systeme am Beispiel eines Flugzeugsystems
In: VDI-Berichte 1842
Düsseldorf: VDI-Verlag, 2004, S. 149–157
- [147] Schäppi, B. Handbuch Produktentwicklung
München: Hanser, 2005
- [148] Scheerer, F. Time-Triggered Technology for Modern Wide Body Aircraft Cabin Pressure Control Systems
In: Open Forum on Time-Triggered Technologies
Toulouse: 24.–25.04.2007
- [149] Schönung, B. E. Numerische Strömungsmechanik
Inkompressible Strömungen mit komplexen Berandungen
Berlin u. a.: Springer-Verlag, 1990
ISBN: 3-540-53137-8
- [150] Schramm, G. Standardisierte Bereitstellung von Simulationsdaten mittels XML
Diplomarbeit
Wedel: FH Wedel, 2007
- [151] Schulz, G. Regelungstechnik
Grundlagern, Analyse und Entwurf von Regelkreisen, rechnergestützte Methoden
Springer-Verlag, 1995
ISBN: 3-540-59326-8
- [152] Schwarz, P. Modellierung und Simulation heterogener technischer Systeme
Arbeitstagung „Schaltungs- und Systementwurf“ DASS'98
Dresden, 1998
- [153] Seemann, J. Software-Entwurf mit UML 2: Objektorientierte Modellierung mit Beispielen in Java
Ausgabe: 2. Auflage.
Berlin, Heidelberg: Springer-Verlag, 2006
ISBN: 978-3-540-30950-5
- [154] Seiffert, U.; Rainer, G. Virtuelle Produktentstehung für Fahrzeuge und Antriebe im Kfz
1. Auflage
Wiesbaden: Vieweg + Teubner Verlag, 2008
ISBN: 978-3-8348-0345-0
- [155] SISC IEEE Std 1516-2000: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules
New York: IEEE Computer Society, 2000
ISBN: 0-7381-2619-5
- [156] Society of Automotive Engineers ARP4754: Certification Considerations for Highly-Integrated Or Complex Aircraft Systems
Society of Automotive Engineers, Inc. (SAE)
Printed in U.S.A., 1996
- [157] Stark, R. CAx Technology to Drive Prototypes in Product Development and Manufacturing
In: VDI-Berichte 1569

Literaturverzeichnis

- [158] Steiger, W. Düsseldorf: VDI-Berichte 2000, S. 267-284
2. Generation PEM-Brennstoffzellen – Erste Erfahrungen
Wolfsburg: Volkswagen AG, 2007
- [159] Strassburger, S. Distributed Simulation Based on the High Level Architecture in Civilian Application Domains
Magdeburg: Otto-von-Guericke-Universität Magdeburg, Dissertation, 2000
- [160] Straßburger, S. Distributed Simulation Based on the High Level Architecture In: Civilian Application Domains
Magdeburg: Otto-von-Guericke-Universität Magdeburg, Dissertation, 2000
- [161] Swink, M. L.; Sandvig, J. C.; Mabert, V. A. Customizing Concurrent Engineering Processes: Five Case Studies
In: Journal of Product Innovation Management, Band 13, Heft 3 1996, S. 229–244
- [162] The HDF Group Hierarchical Data Format
[online]. Copyright: -, letzte Aktualisierung: 06.07.2007, erhältlich im Internet unter
<http://www.hdfgroup.org/>
- [163] The MathWorks Using Simulink
Simulation and Model-Based Design
Version 6
The MathWorks, 2004
- [164] The Object Management Group The Object Management Group (OMG)
[online]. Copyright: 1997–2007, letzte Aktualisierung 13.07.2007, erhältlich im Internet unter
<http://www.omg.org/>
- [165] The Object Management Group SysML – Open Source Specification Project
[online]. Copyright: -, letzte Aktualisierung: 18.07.2007, erhältlich im Internet unter
<http://www.sysml.org/>
- [166] Thomas, S. SSL & TLS essentials : securing the Web
New York, NY [u. a.]: Wiley, 2000
ISBN: 0-471-38354-6
- [167] Tiller, M. Introduction to physical modeling with Modelica
2. Auflage
Boston: Kluwer Acad. Publ., 2004
ISBN: 0-7923-7367-7
- [168] Tueber, K. Analyse des Betriebsverhaltens von Polymer-Elektrolyt-Membran-Brennstoffzellen für portable Systeme
Essen: Universität, 2004
- [169] U.S. Department of Transportation Advisory Circular, Subject: Fuel Tank Flammability, Initiated by: ANM-112, AC No: 25.981-2A
Federal Aviation Administration
- [170] Ullmann, M. Integrationsuntersuchung eines Brennstoffzellen-Hilfsenergiesystems an Bord eines zukünftigen Verkehrsflugzeuges unter Verwendung moderner Simulationstechniken
Diplomarbeit
Hamburg: HAW, 2007
- [171] Ullmann, M. Integrationsuntersuchungen eines Brennstoffzellen-Hilfsenergiesystems an Bord eines zukünftigen Verkehrsflugzeuges unter Verwendung moderner Simulationstechniken
Hamburg: Hochschule für angewandte Wissenschaften, 2007
- [172] Ulrich, K. T.; Eppinger, S. D. Product Design and Development
3. Auflage
New York: McGraw-Hill/Irwin, 2004
- [173] Vajna, S. Die neue Richtlinie VDI 2219: Praxiserprobte Hinweise zu Einführungsstrategien und Wirtschaftlichkeit von EDM/PDM-

Literaturverzeichnis

- Systemen
In: VDI-Bericht 1497 Beschleunigung der Produktionsentwicklung durch EDM/PDM- und Feature-Technologie, 1999, S. 25–42
ISBN: 3-18-091497-1
- [174] Vajna, S. et al. The Autogenetic Design Theory: an evolutionary view of the design process
In: Journal of Engineering Design Vol. 16, No. 4, August 2005, S.423–440
- [175] Vajna, S. et al. FEA Systems as Evaluation Tool to Optimise Products with Evolutionary Algorithms
In: NAFEMS The international Association for the Engineering Analysis Community, Seminar: Integration of Numerical Simulation into the Development Process, 17.–18. November 2003, S. 1–8
- [176] VDI VDI Protokoll: VDI-Erfa-Kreis 2 „Hierarchische Simulationsmodelle" Dortmund: 1998
- [177] Verein Deutscher Ingenieure VDI- Richtlinie 3633: Simulation von Logistik-, Materialfluss- und Produktionssystemen – Grundlagen
Hrsg.: Deutsches Institut für Normung
Berlin: Beuth Verlag, 2000
- [178] Verein Deutscher Ingenieure VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte
Hrsg.: Deutsches Institut für Normung
Berlin: Beuth Verlag, 1993
- [179] Verein Deutscher Ingenieure VDI-Richtlinie 2206: Entwicklungsmethodik für mechatronische Systeme
Hrsg.: Deutsches Institut für Normung
Berlin: Beuth Verlag, 2004
- [180] Verein Deutscher Ingenieure VDI-Berichte 1900 Erprobung und Simulation in der Fahrzeugentwicklung – Mess- und Versuchstechnik –
Düsseldorf: VDI-Verl., 2005
ISBN: 3-18-091900-0
- [181] Verein Deutscher Ingenieure VDI-Berichte 1967.1 Berechnung und Simulation im Fahrzeugbau
Düsseldorf: VDI-Verl., 2006
ISBN: 3-18-091967-1
- [182] Verein Deutscher Ingenieure VDI-Richtlinie 2219 Informationsverarbeitung in der Produktentwicklung – Einführung und Wirtschaftlichkeit von EDM/PDM-Systemen
Hrsg.: Deutsches Institut für Normung
Berlin: Beuth Verlag,
- [183] Wallaschek, J. Modellierung und Simulation als Beitrag zur Verkürzung der Entwicklungszeiten mechatronischer Produkte
In: VDI-Bericht 1215 Simulation in der Praxis – Neue Produkte effizienter Entwickeln, 1995, S. 35–50
ISBN: 3-18-091215-4
- [184] Watkins, D. S. Fundamentals of matrix computations
Second Edition
New York, NY: Wiley-Interscience, 2002
ISBN: 0-471-21394-2
- [185] Wegner, B. Autogenetische Konstruktionstheorie – Ein Beitrag für eine erweiterte Konstruktionstheorie auf der Basis Evolutionärer Algorithmen
Magdeburg: Fakultät für Maschinbau, 1999
- [186] Weilkiens, T. Systems Engineering mit SysML/UML
Modellierung, Analyse, Design
Dpunkt.Verlag, 2006
ISBN: 3-89864-409-x
- [187] Wendt, W. Taschenbuch der Regelungstechnik
5. Auflage
Frankfurt am Main: Deutsch, 2003
ISBN: 3-8171-1705-1

Literaturverzeichnis

- [188] Wiggers, N. Konzeption eines integrierten Prozessmodells für die systemübergreifende, simulationsgestützte Entwicklung von Kabinensystemen
Diplomarbeit
Hamburg: HWI TH, 2006
- [189] Winkler, W. Brennstoffzellenanlagen
Heidelberg: Springer-Verlag, 2002
ISBN: 3-540-42832-1
- [190] World Wide Web Consortium (W3C) Extensible Markup Language (XML) 1.0 (Fourth Edition)
[online]. Copyright: 2006, letzte Aktualisierung: 18.07.2007, erhältlich im Internet unter <http://www.w3.org/TR/REC-xml/>
- [191] Zeigler, B. et al. Theory of Modeling and Simulation Integrating Discrete Event and Continuous Complex Dynamic Systems
Second Edition
San Diego: Calif. u.a: Academic Press, 2000
ISBN: 0-12-778455-1
- [192] Zorriassatine, F.; Wykes, C.; Parkin, R.; Gindy, N. A survey of virtual prototyping techniques for mechanical product development
In: Proceedings of the Institution of Mechanical Engineers, Part B (Journal of Engineering Manufacture), Band 217, Heft 4, 2003, S. 513–530
- [193] Zuber, E.; Kress, M.; Wagner, W. Virtuelle Produktion – Partner der digitalen Produktentwicklung
In: VDI-Berichte 1569
Düsseldorf: VDI-Verlag, 2000, S. 249–266

Stichwortverzeichnis

A

Abstraktionsgrad 13, 69, 76
Adam-Bashforth 140
Adam-Moulton 140
Aktivierungsverluste 85
algebraische Schleife 142
alleinstehende Simulation 44, 46, 53, 81
analoge Simulation 4
Anforderungsdiagramm 161
Anode 84, 86, 105
Architekturebene 13, 66, 68
ATA-Kapitel 26, 29, 66, 68, 72, 73, 133, 134
autotherme Reformierung 87
Auxiliary Power Unit IV, 90, 94

B

BDF 140
Blockdiagramm 78, 161
Blockschaltbilder 150, 151
Bottom-up 7, 8
Brennstoffzelle VIII, 84, 85, 86, 90, 91, 92, 94,
95, 96, 97, 98, 99, 100, 101, 102, 103, 104,
105, 106, 107, 108, 109, 110, 111, 112, 113,
115, 118, 120, 121, 122, 123, 124
Butler-Volmer-Gleichung 86, 109

C

Carnot-Wirkungsgrad 91, 103
Client-Server 23, 24, 131, 155
Comma Separated Values 158
Computational Fluid Dynamics 145, 146
Computer Aided Engineering 20
Concurrent Versions System 155
CORBA 16, 120

D

Dampfpreformierung 87, 105, 118
Datenbankmanagementsystem 127, 128
Datenmanagementsystem 154, 159
DC-Analyse 153
Detaillierungsebene 9, 45, 50, 107
Detaillierungsphase 8, 11, 27, 32, 33, 38, 43,
44, 46, 50, 51, 58, 59, 68, 69, 73, 81, 82,
107, 111, 125, 137, 159
Differenzialgleichung 139, 140, 148
digitale Simulation 4
Document Object Model 133
Dreiecksmatrix 139
Druckspeicherung 86
Durchtrittsspannung 86

E

Edukte 84
Einschrittverfahren 140
endotherm 87, 88
Energiebilanz 147

Energiewandlung 84, 96
Engineering Data Management 23, 29
Entwicklungsdokumente 23, 27, 65
Erhaltungssatz 145
Euler-Verfahren 145
exotherm 84, 87
explizite Form 140
Extensible Markup Language 126, 157

F

Federate 120, 122, 123, 124, 125
Federation Execution 120, 125
feste Simulationskopplung 16, 119, 138
Finite-Differenzen 144, 145
Finite-Elemente 17, 144
Finite-Volumen 144, 145
Fischer-Tropsch-Kraftstoff 92
Flottenemission 90
Funktionsebene 66

G

gaußsche Eliminationsverfahren 139
Gleichgewichtspunkte 142
Grenzschicht 85

H

Hardware in the Loop 11, 17, 18, 30, 38, 46
Hauptsystemanforderungen 13, 27, 31, 34, 44,
50, 65, 80, 82
Hierarchical Data Format 126, 156
High Level Architecture 16, 120
Hochtemperaturbrennstoffzelle 98
Hybridantrieb 90

I

implizite Form 140
Inertisierung 103, 106
Informationsbereitstellung 154
Informationsmanagement 25, 32, 37, 38, 40,
41, 44, 45, 46, 48, 60, 63, 75, 76, 83, 125
Informationsvisualisierung 154
Integrationsmethode 139

J

Jacobi-Matrix VII, 140, 141, 142, 153
Java Database Connectivity 127

K

Kathode 84, 86
Kirchhoffsche Knotenanalyse 153
Kohlenwasserstoff 87, 88, 89, 92, 104
Komponentenebene 13, 26, 27, 81
Komponentenentwicklung 11, 13, 26, 30, 32,
33, 35, 45, 46, 50, 51, 65, 68, 69, 72, 73, 81,
82, 107, 119
Konfigurationsmanagement 23, 24, 41, 62, 76

Stichwortverzeichnis

Kontinuitätsgleichung 146
 Konzeptentwicklung 9, 11, 12, 19, 26, 31, 37, 38, 46, 48
 Konzeptphase 6, 9, 18, 21, 27, 30, 31, 34, 39, 43, 44, 45, 47, 48, 51, 58, 65, 68, 69, 72, 73, 80, 81, 82, 95, 106, 119, 137, 159
 Koordinationsfunktion 45, 46, 50, 51, 52, 53, 55, 57, 59, 60, 61, 62, 67, 70, 74, 75, 76, 81
 Kurzschlussbereich 99

L

Ladungsträger 85, 86, 109
 Laplace-Transformation 139
 Leerlaufspannung VIII, 86, 96, 107, 108, 109
 lose Simulationskopplung 25, 68, 114, 126, 136, 138
 Luftüberschusszahl VIII, 102, 103

M

Massentransportverluste 86
 mathematisch orientierte Simulation 133, 148, 149
 Mechatronik 8, 34
 Mehrschrittverfahren 140
 Mensch-Maschine-Schnittstelle 154
 Metainformation 72, 133
 Metallhydridspeicherung 86
 Modellbibliotheken 56, 59, 148
 Modelldokumentation 66, 67, 68, 69, 80
 Modellimport 15, 55
 Modellwiederverwendung 41, 49, 59, 68
 Modularisierung 41, 56, 59, 95, 107
 More Electrical Aircraft 93
 Multidisziplinäre Simulation 59

N

Navier-Stokes 145, 146, 147
 nernstschen Gleichung 86, 108
 Netzwerksimulator 151
 Newton-Verfahren 141, 142, 153
 numerische Methoden 3

O

Oxidationsreaktion 84, 88

P

Paladium 88
 partielle Dehydrogenierung 88
 partielle Oxidation 87
 PEMFC 84, 85, 91, 94, 96, 101, 104, 105, 106, 107, 118
 Plasmareformierung 89
 Polarisationskennlinie 84, 107, 109
 pq-Formel 141
 Primärenergieträger 92, 94
 Problemlösezyklus 7, 8, 9
 Produktdatenmanagement 23
 Produktentstehungsprozess 2, 8, 23, 26, 34, 43

R

Ram Air Turbine 95
 Reaktionsgleichung 84, 88
 Redox-Reaktion 108
 Reformierung 87, 89, 103, 104, 105, 106, 109, 111
 Runge-Kutta-Verfahren 140
 Run-Time Infrastructure 120

S

Schnittstellenarchitektur 55, 56
 Schnittstellendokumente 55, 67
 Sekundärenergie 92, 93
 Separator 88, 98
 Sequenzdiagramm 71
 signalflussorientierte Simulation 149
 Simulationskopplung 13, 15, 16, 25, 44, 45, 55, 68, 76, 81, 114, 119, 120, 125, 126, 137, 138
 Simultaneous Engineering 19
 SOFC 92
 Standardreaktionsenthalpie 84, 88
 Standardumgebungsbedingungen 103
 Stromdichte VII, 85, 86, 96, 97, 100, 102, 104
 Strömungsfeld 146
 symbolische Mathematik 148, 149
 symbolorientierte Simulation 5, 150
 SysML 65, 78, 79, 80, 83, 134, 135, 137, 138, 161, 162
 System Modeling Language 78
 Systemanforderungen 31, 48, 50, 65, 66, 81, 82
 Systemarchitektur 10, 27, 31, 34, 37, 45, 51, 65, 81, 117
 Systembeschreibungssprache 35, 37
 Systemdekomposition 13, 31
 Systementwicklung 2, 3, 4, 9, 19, 26, 28, 30, 31, 34, 35, 36, 39, 45, 48, 65, 66, 79, 80, 84, 107, 111, 120, 144
 Systementwicklungsprozess 2, 6, 10, 26, 34, 36, 38, 80, 111, 120
 Systeminteraktion 27
 Systems Engineering 53, 77
 Systemsimulation 5, 18, 21, 32, 58, 77, 110, 111, 113, 115, 117, 148

T

Taylor-Reihe 139, 140, 141, 145
 Testumgebung 16, 33, 34, 45
 Top-down 7, 8, 11, 13, 26, 27, 31, 95, 107
 Transaktionsserver 128, 130, 131, 133
 transiente Analyse 153
 Turbulenzmodelle 147

U

UML 77, 78

V

Validierung 4, 8, 9, 10, 11, 21, 27, 31, 32, 41, 51, 53, 59, 65, 73, 81, 82, 127
 Verhaltensebene 69

Stichwortverzeichnis

Verifikation 8, 9, 11, 51, 53, 58, 65, 66, 70, 80,
81, 82, 83, 137
Versionsverwaltungssystem 155, 156
verteilte Simulation 16, 55, 119, 120, 121, 122,
124, 125, 138
virtuelle Verifikation 11, 65, 66, 80, 83, 137
V-Modell 7, 8, 9, 10, 11, 13, 26, 27, 28, 33, 37,
47, 50, 51, 58, 65, 80, 82, 83, 95, 107, 137
Vorgehensmodell 6, 7, 8, 9, 21
VV&A 53

W

Wärmekraftmaschine 109
Wärmerückgewinnung 103, 106
Wasserstoffspeicherung 86, 91
Web-Frontend 131, 154

X

XML Metadata Interchange 158

Z

zeitdiskret 4, 18
Zeitmanagement 16, 120, 122
Zellspannung 86, 96, 99, 100, 104, 105, 108,
109, 110
zentrale Datenbank 61, 64
zentrale Informationsbasis 62, 64, 138
Zugriffsrechte 28, 36, 37, 39, 61, 62, 67, 72,
73, 75, 76, 136, 138
Zustandsautomat 78, 135
Zustandsgleichung 139, 147